Improving Learning Experience in MOOCs with Educational Content Linking

by

Shang-Wen Li

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
February 2017

© Massachusetts Institute of Technology 2017. All rights reserved.

Signature redacted

Author .......... 
Department of Electrical Engineering and Computer Science 
January 31, 2017

Signature redacted

Certified by ..........
Victor W. Zue 
Professor of Electrical Engineering and Computer Science 
Thesis Supervisor

Signature redacted

Accepted by ..........
Leslie A. Kolodziejski 
Chair, Department Committee on Graduate Students
Improving Learning Experience in MOOCs with Educational Content Linking

by

Shang-Wen Li

Submitted to the Department of Electrical Engineering and Computer Science on January 31, 2017, in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

Since the first MOOC (Massive Open Online Course) in 2011, there have been over 4,000 MOOCs on various subjects on the Web, serving over 35 million learners. MOOCs have shown the ability to transcend time and space, democratize knowledge dissemination, and bring the best education in the world to every learner. However, the disparate distances between participants, the size of the learner population, and the heterogeneity of the learner backgrounds make it difficult for instructors to interact with learners in a timely manner, which adversely affects their learning outcome.

To address these challenges, in this thesis, we propose a framework of educational content linking. By linking pieces of learning content scattered in the various course materials into an easily accessible structure, we hypothesize that this framework will guide learners and improve content navigation. Since most instruction and knowledge acquisition in MOOCs takes place when learners are surveying course materials, better content navigation may help learners find supporting information to clear up confusion and improve the learning outcome.

To support our conjecture, we present end-to-end studies to investigate our framework around two research questions. We first ask, does manually generated linking improve learning? To investigate this question, we choose two STEM courses, statistics and programming language, and demonstrate how the annotation of linking among course materials can be accomplished with collaboration between course staff and online workers. With this annotation, we implement an interface that can simultaneously present learning materials and visualize the linking among them. In a large-scale user study, we observe that this interface enables users to find desired course materials more efficiently, and retain more concepts more readily. This result supports the notion that manual linking does indeed improve learning outcomes. Second, we ask, can learning content be generated using machine learning methods? For this question, we propose an automatic linking algorithm based on conditional random fields. We demonstrate that automatically generated linking can still lead to better learning, although the magnitude of the improvement over the unlinked interface is smaller. We conclude that the proposed linking framework can be implemented at
scale with machine learning techniques.

Thesis Supervisor: Victor W. Zue
Title: Professor of Electrical Engineering and Computer Science
Acknowledgments

I would never have been able to finish my dissertation without the support from my advisor, committee members, colleagues, friends, and family. I would like to thank:

- My advisor Victor for inspiring me with his passion and intellectual curiosity, and guiding me with encouragement and patience. His care for education and computation for humanity has taught me that the meaning of research is to change the world and make our society better.

- My committee Jim for his insightful suggestions on my work and his organization of the SLS group. He provides an excellent atmosphere for doing research.

- My committee Rob for his sharp comments. His feedback on the design of the user research and the interfaces greatly improved this thesis.

- Stephanie for her feedback on this thesis about my work and the writing. She significantly enhanced the way I frame my research.

- Piotr for offering a chance for internship at edX. Working with him has greatly expanded my view on the education field.

- all SLS members including Ann, David, Di-Chia, Ekapol, Hongyin, Jacqueline, Jennifer, Jingjing, Mandy, Marcia, Maryam, Michael, Mitra, Najim, Scott, Sree, Stephen, Tuka, Wei-Ning, Xue, Yonatan, and Yu. They are wonderful folks who always inspire me in both research and life. It is my pleasure to work with them.

- Carrie, Chen, Chengjie, Hung-Yi, Juho, and Xiangrong for the collaboration and for exploring this exciting research field with me.

- Alice, Amy, Benson, Ching-Feng, Chiyuan, Darren, Junan, Kun-Hua, Peng, Tianfan, Wenzhen, Xuhong, Yu-Chung, and Zhendong for sharing all the good and bad times in life with me and keeping me sane.

- My friends and family in both the U.S. and Taiwan for their love and support.
• My parents for their unconditional love and not forgetting me even though I call home only every four months.

• Crystal for her company and guidance. She was the light at the end of the PhD tunnel, and led me through the thorns in life.

• 6.00x course staff and the thousands of Turkers for helping me with my user research. Without their contribution, this thesis would not have been possible.

This work is supported by Quanta Computer.
Contents

1 Introduction 21
  1.1 Motivation ................................................. 23
  1.2 Educational content linking ............................. 24
  1.3 Contributions ............................................. 27
  1.4 Thesis overview ......................................... 28

2 Background 29
  2.1 Learning science ......................................... 29
  2.1.1 Tutoring at scale .................................... 30
  2.1.2 Course navigation ................................... 32
  2.2 Crowdsourcing ............................................ 33
  2.2.1 Micropayment workforces ............................ 34
  2.2.2 Quality control ...................................... 35
  2.3 Machine learning and human language understanding 37
  2.3.1 Human language technology ........................ 37
  2.3.2 Conditional random fields (CRF) .................... 39
  2.3.3 Word embedding ...................................... 43
  2.4 Corpora .................................................. 46
  2.4.1 Course subjects ...................................... 46
  2.4.2 Course materials ...................................... 48

3 Would linking help learning? 51
  3.1 Linking materials ....................................... 52
3.1.1 The linking tree ............................................. 52
3.1.2 Homologous and heterologous linking ..................... 53
3.1.3 Linking representation ..................................... 56
3.1.4 Annotation tasks ............................................. 58
3.2 Presenting linking to learners .................................. 63
3.3 Comparative study ............................................. 68
  3.3.1 Study design ................................................ 69
  3.3.2 Baseline ..................................................... 71
  3.3.3 Experiment subjects ....................................... 72
  3.3.4 Experiment scale .......................................... 74
3.4 Results .......................................................... 75
  3.4.1 How linking affects search ................................ 76
  3.4.2 How linking affects information memorization ............. 85
3.5 Click log analysis ............................................... 90
3.6 Conclusions ..................................................... 95

4 Can we link automatically? ........................................ 97
  4.1 Problem formulation .......................................... 99
  4.2 Sequential tagging with CRF .................................. 101
  4.3 Feature extraction ............................................ 104
  4.4 Evaluation: similarity to human labeling ...................... 110
  4.5 Evaluation: benefit in learning ............................... 118
    4.5.1 How automatic linking affects search ...................... 119
    4.5.2 How automatic linking affects information memorization ... 123
  4.6 Difference pattern analysis .................................... 127
  4.7 Comparison with the edx interface ............................ 133
    4.7.1 The edx interface ......................................... 133
    4.7.2 User study, results, and discussions ...................... 135
  4.8 Conclusions ................................................... 139
5 Conclusions

5.1 Summary and contributions .............................................. 141
5.2 Future work ................................................................. 143
  5.2.1 Learning platforms of the future ................................... 143
  5.2.2 Towards a variety of course subjects and material types .... 144
  5.2.3 More sophisticated algorithm for linking at scale .......... 144

A Sampled problems and topics for the user study .................. 147
  A.1 Sampled problems ..................................................... 147
    A.1.1 Problems of Stat2.1x .............................................. 147
    A.1.2 Problems of 6.00x ................................................ 149
  A.2 Sampled topics .......................................................... 151
    A.2.1 Topics of Stat2.1x ................................................ 151
    A.2.2 Topics of 6.00x .................................................... 152
## List of Figures

1-1 Schematics of the transformation of several independent course materials to a linked structure. Each color illustrates one type of education material. Note that here linking refers to conceptual relations across materials. .................................................. 25

1-2 Experimental pipeline for the question *Would it help learners if we are able to link course materials using human annotators?* .......................... 26

1-3 Experimental pipeline for the question *Can the courseware be linked at scale using machine learning methods?* ........................................... 26

2-1 Diagram of general CRFs and linear chain CRFs. .............................. 40

2-2 Architecture of feedforward neural network employed to obtain word2vec embedding. ................................................................. 45

3-1 Trunk-and-leaves architecture (i.e., the linking tree) .......................... 53

3-2 Examples of homologous (upper panel) and heterologous linking (lower panel) .............................................................. 54

3-3 Annotating homologous linking as an alignment problem .................... 56

3-4 Annotating heterologous linking as either an alignment problem (upper panel), or a binary classification problem (lower panel) .................. 57

3-5 Website built to collect homologous linking (i.e., the alignment between lecture video transcription and slides) ................................. 59

3-6 Website built to collect heterologous linking (i.e., the binary classification task of deciding whether a video vignette/discussion thread pair or a video vignette/textbook section pair is relevant) ....................... 60
3-7 The implemented interface which presents learning content and linking information simultaneously. In this interface, course materials are retrieved by submitting queries using the provided search tool. The retrieved materials are listed according to their types (in the top left corner of this screenshot) and their original positions in each material sequence (e.g., lecture or chapter indices). Titles of listed materials are shown on the left hand side, and content selected from the list is in the middle. If the selected content is a video (i.e., the trunk), linked supplementary objects (i.e., the leaves) are also displayed as orange, green, or pink blocks under the video scrubber.

3-8 By clicking on any of the colored blocks under video scrubber, content of the linked supplementary object represented by this clicked block is rendered in a lightbox. In this figure, we provide an example illustrating how a linked slide is displayed in the proposed interface after its corresponding block is clicked.

3-9 With the annotated linking from video segments (i.e., sentences or vignettes) to slides, textbook, and discussions, as well as the time code of each segment, the synchronized linked objects under the video scrubber can be rendered. In this example, each page of slides is indexed with A, B, and C; each textbook section is indexed with α, β, and so on; each discussion thread is indexed with a, b, c, and so on.

3-10 The first row shows two sampled problems used in the information search learning scenario. For each of the problems, shown in the second row is a piece of learning content that is accepted as the answer; a textbook section (content titled "The Range, IQR and SD") and a page of slides (titled "Types of Exceptions") is displayed respectively. In this figure, the left hand side is a problem-answer pair for Stat2.1x; the right is from 6.00x.
3-11 The first row shows two sampled topics used in the concept retention scenario. For each topic, shown as an example is an essay submitted by a learner in our user study. We also set concepts in essays in bold font. In this figure, the left hand side is a topic-essay pair for Stat2.1x and the right is for 6.00x.

3-12 The null interface, which serves as one of our baselines. This interface retains linking's components for key-term search, material list, and content presentation. The layout and visual design are also identical. The only difference is that we strip away the linking visualization, and there is no synchronized supplementary learning object under each lecture video (i.e., the trunk).

3-13 Improvement in search time (upper panel) and accuracy (lower panel) when linking interface is used. Learning performance improvement was measured in the Stat2.1x study. Also shown are the 95% confidence intervals (shown as error bars) and significance test results (statistically significant differences marked with red asterisk).

3-14 Improvements in search time and accuracy when linking interface was deployed. Learning performance was measured in the 6.00x study. The 95% confidence intervals and significance test results are also plotted.

3-15 This first-order regression model (dashed line) relates the average search time (in seconds) of task batches (horizontal axis) to the accuracy improvement yielded by deploying the linking interface (vertical axis).

3-16 The improvement in the number of unique key-terms contained in submitted essays when linking interface was used. Learning performance was measured in the Stat2.1x study. The 95% confidence intervals and significance test results are also provided.

3-17 The improvement in the number of unique key-terms when the linking interface was deployed. Learning performance was measured in the 6.00x study. The 95% confidence intervals and significance test results are also provided.
3-18 Question asked in the tasks where we recorded the two sampled search paths.

3-19 The sampled search path recorded when a subject used the null interface to complete the assigned task. In this path this subject surveyed ten objects, three of which contained valid learning pieces. The three objects are indicated with their titles in red; the titles of the remaining objects are set in cyan. These numbers equal the average of the null interface.

3-20 The sampled search path recorded when a subject used the linking interface to complete the same task as in Fig. 3-19. In this path this learner surveyed seven objects, three of which contained valid learning pieces. These numbers equal the average of the linking interface.

4-1 The same linking configuration represented with two different label sets. In the upper panel, the original label set described in Section 4.2 is used. A, B, and C denote the aligned slide index; Y and N denote whether the considered object is linked. In the lower panel, the original label set along with a boundary label (denoted as 'bnd') is used.

4-2 The improvement in search time (upper panel) and accuracy (lower panel) with the linking (red bars) or auto linking (black bars) interface, with the null interface as the baseline. Learning performance improvement was measured in the Stat2.1x study. The 95% confidence intervals (shown as error bars) and significance test results (marked with red asterisk if the difference is statistically significant) are also provided.

4-3 Improvements in search time and accuracy when different interfaces were used in the 6.00x study. This plots the improvement yielded when using linking (red bars) and auto linking (black bars) as compared to null.
4-4 The improvement in the number of unique key-terms contained by submitted essays when the *linking* (red bars) or *auto linking* (black bars) interface was used, with the *null* interface as the baseline. Learning performance was measured in the Stat2.1x study. The 95% confidence intervals and significance test results are also provided.

4-5 The improvement in the number of key-terms contained in submitted essays when different interfaces were deployed in the 6.00x study. Also visualized is the improvement from using *linking* (red bars) and *auto linking* (black bars) as compared to *null*.

4-6 An example of difference pattern 1. The left panel shows a sampled discussion thread; the right panel presents the vignette linked to the thread by human annotators (upper right) and by the proposed CRF algorithm (lower right). The case where none of the vignettes is linked is represented by $\emptyset$.

4-7 An example of difference pattern 2. The left panel shows a sampled discussion thread; the right panel presents the vignette linked to the thread by human annotators (upper right) and the proposed algorithm (lower right).

4-8 An example of difference pattern 3. In this example, the thread (left panel) was linked by humans and machine to two vignettes (right panel) in the same lecture video. The two vignettes are closely related to each other, and the difference in presenting these two ways of linking was only minor.

4-9 An example of difference pattern 4. Here, humans and machine linked two vignettes (right panel) from various lecture videos to the same discussion the thread (left panel).
4-10 The implemented edx interface that reproduced the design and layout of the edX website to offer learners a user experience similar to that of a real MOOC, except that we added the additional search tool to access course materials. This interface was used to investigate how much added value our linking framework provides for state-of-the-art MOOC platforms. .................................................. 134

4-11 Improvement in search time and accuracy for different interfaces. Plotted are the improvement from using linking (red bars) and auto linking (black bars) as compared to null, and the improvement from linking (blue bars) and auto linking (orange bars) as compared to edx. ......... 137

4-12 The improvement in the number of unique key-terms contained by submitted essays when different interfaces were deployed. The bars are pictured as in Fig. 4-11. ............................................... 138
## List of Tables

2.1 Summarization of sizes of course materials used in this thesis. 49

3.1 Sizes of comparative studies on Stat2.1x and 6.00x 75

3.2 Number of tasks completed by each cohort for each learning scenario (i.e., information search and concept retention) and interface (i.e., *null* and *linking*) in the Stat2.1x study 76

3.3 Number of tasks completed by each cohort for each learning scenario (i.e., information search and concept retention) and interface (i.e., *null* and *linking*) in the 6.00x study 76

3.4 Learner performance in information search scenario in Stat2.1x study. Performance is evaluated by the average search time and average accuracy metrics, and measured within various cohorts using different interfaces. 77

3.5 Learner performance in the information search scenario in the 6.00x study. Similar to the first study, performance is evaluated by the average search time and average accuracy, and measured in various cohorts using different interfaces. 81

3.6 Learner performance in the concept retention scenario in the Stat2.1x study. Performance was evaluated by the number of unique key-terms in submitted essays and measured within various cohorts using different interfaces. 86
3.7 Learner performance in the concept retention scenario in the 6.00x study. As with the Stat2.1x study, performance was evaluated by the number of unique key-terms in the submitted essays and measured within various cohorts using different interfaces. 89

3.8 In the information search and concept retention scenario of the 6.00x study, we computed the three metrics (number of search queries used to accomplish a task, number of learning objects surveyed in each task, and the spent time in each learning object) for the two interfaces. The averages ($\mu$) and standard deviations ($\sigma$) of the three metrics are listed here. 91

4.1 The F1 scores (%) of automated linking systems in Stat2.1x using different models (logistic regression and CRF) as well as lexical (BoW stands for bag-of-words and word2vec for the neural network word embedding) and visual (HSV=HSV histogram and HP=horizontal projection) features. Performance of both homologous (i.e., linking between video sentences and slides) and heterologous (i.e., linking between video vignettes and textbook sections) tasks is listed. In the table, the parentheses after word2vec denote that the HP visual features were deployed only in the homologous task. 111

4.2 F1 scores (%) of automated linking systems in 6.00x using different models (logistic regression and CRF) and lexical (BoW and word2vec) and visual (HP) features. Performance of both homologous (i.e., linking between video sentences and slides) and heterologous (i.e., linking between video vignettes and textbook sections/discussion threads) tasks is listed. 115

4.3 Best performance of proposed automated linking system (evaluated with F1 scores, listed in first two rows) and annotator agreement (evaluated with kappa scores, listed in third and fourth rows) in each linking task 117
4.4 Learner performance in the information search scenario of the Stat2.1x study. Performance was evaluated by the average search time and average accuracy metrics, and measured within various cohorts using the null, linking, and auto linking interfaces. ........................................ 120

4.5 Learner performance in the information search scenario in the 6.00x study. Performance was evaluated by the average search time and average accuracy metrics, and measured within various cohorts using the null, linking, or auto linking interfaces. ........................................ 120

4.6 Learner performance in the concept retention scenario in the Stat2.1x study. Performance was evaluated by the number of unique key-terms in submitted essays and measured within various cohorts using the null, linking, and auto linking interfaces. ........................................ 124

4.7 Learner performance in the concept retention scenario in the 6.00x study. Performance was evaluated by the number of unique key-terms in submitted essays and measured within various cohorts using the null, linking, and auto linking interfaces. ........................................ 126

4.8 The number of threads categorized into the four difference patterns. 128

4.9 Learner performance in the information search scenario in the 6.00x study. In addition to the results reported in Table 4.5, we list performance (evaluated by the average search time and average accuracy) measured when the edx interface was used. ........................................ 136

4.10 Learner performance in the concept retention scenario in the 6.00x study. In addition to the results reported in Table 4.7, we list the number of unique key-terms in submitted essays measured when the edx interface was used. ........................................ 136
Chapter 1

Introduction

In 2011, a MOOC (massive open online course) revolution began in university education [71, 99]. Today, a mere five years after the first MOOC was launched, over 4000 MOOCs, from science and engineering to humanities and law, have been offered on the Web and have served over 35 million learners on platforms such as Coursera, edX, Udacity, and FutureLearn [126, 27, 32, 128, 38]. These MOOCs have been created by over 500 of the world's top institutions and have been taught by the top instructors. In addition, MOOCs allow free enrollment and enable learners around the globe to take courses without the need for physical presence. Thus, MOOCs have the potential to transcend time and space, democratize knowledge dissemination, and bring opportunities to learners in every corner of the world.

MOOCs inspire a new model in the delivery of quality education. In conventional residential education, classes have much smaller sizes. These classes are taught in thousands of institutions on the same subject with only slight variations. In contrast, MOOCs adopt a distributed model. This model can accumulate the investment of offering these classes in institutions and instructors, and allow course builders to allocate their time and effort more efficiently in implementing in each course various state-of-the-art and research-based pedagogies such as active learning, mastery learning, and cooperative learning [97, 29, 28]. Thus, MOOCs provide enormous educational value to learners and instructors. Evidence suggests that well-designed MOOCs alone can lead to high levels of student learning and satisfaction [97]. In addition,
we have observed a growing trend of college instructors adopting this approach in blended classrooms [78]. In blended learning, residential classroom instructors utilize existing MOOC content to save effort in course material preparation, and thus leave time to focus on interacting with students to create a learner-centered environment [114, 25, 50].

However, the open and free character of MOOCs has also created a set of challenges that are not observed in conventional education, that is, the sheer size of the learner body, and the heterogeneity of their backgrounds [56]. A MOOC typically has thousands to tens of thousands of learners with various demographics, course preparedness, learning goals, and motivations. Given this class size and heterogeneity, the conventional one-size-fits-all pedagogy is not sufficient. For example, in the same MOOC, some learners may struggle with elementary concepts due to insufficient prerequisite backgrounds, while another group of learners may already have years of experience in the industry of the relevant area and their learning goal is to update their job skills. Due to physical distance, learners in MOOCs usually rely on self-regulated learning to meet their own learning needs. For instance, a learner who is confused about a topic in the lecture video may choose to pause the video, turn to a textbook or discussion forum for a more understandable description, and return to the video when the learner has a better understanding about the underlying topic. In this way, different learners take various learning paths and learning materials for their diverse learning needs. Nonetheless, because of the unfamiliarity of learners with the course subject as well as the amount of learning content in a MOOC, it is usually cumbersome for learners to find suitable content.

To address these challenges, in this thesis we propose an educational content linking framework which allows the linking and organizing of the scattered educational materials in a MOOC, as well as the visualization of the conceptual relations across these materials. Since visualization assists learners in navigating the material, we expect this framework to help learners achieve self-regulated learning by allowing them to find appropriate information efficiently.
1.1 Motivation

One-to-one tutoring has been shown to be extremely effective in enhancing learning outcomes [12]. However, as this approach is too costly, for many years educators have dreamt of achieving similar effectiveness with a scalable approach [12]. A significant number of studies have attempted to determine how people learn and why one-to-one tutoring is so successful in improving learning [129, 46]. One of the key factors could be constructive struggle: much research have shown that keeping learners in a state of engagement between boredom and confusion has a substantial positive impact on learning [35, 115, 110]. Outside the laboratory, this strategy has also been commonly applied to keep learners engaged, e.g., by asking students questions, inserting quizzes into lecture videos, or providing instructional scaffolding (Instructors provide sufficient support to learn a concept, while, during the entire learning process, support is taken away gradually to promote learners developing deeper-level knowledge).

Timely responses to confusion play a crucial role in the success of this strategy, and failing to do so can affect learning in the opposite way, leading to frustration, or causing learners to cease participating. In a MOOC scenario, the incredibly low instructor-to-learner ratio and the heterogeneous background of learners make responding to learning needs extremely challenging. To address the problem, typically instructors can provide pre-defined hints, optional course materials, or even intelligent tutoring systems (ITS) to serve various needs and address sources of confusion. Another alternative is to rely on the learners themselves to discover answers in course forums or on the Web. Although helpful, both approaches also have their downsides. Providing hints, optional materials, or ITS, even with the help of state-of-the-art machine learning methods, involves significant handcrafting, such as designing banks of responses or individualized pathways for different needs. This approach is neither scalable nor generalizable from course to course. Furthermore, application of this approach in undergraduate-level or graduate-level subjects, which are the focus of MOOCs, is more cumbersome, since concepts at such a subject level are much more complicated. In contrast, the alternative is much more scalable. The learnersourcing
model [63] potentially generates responses to diverse learning needs at scale. However, due to the amount of generated responses and needs, matching between the two is challenging. For example, although ideally a learner can clear up every confusing point by using the MOOC forum or the Web, looking for useful contents from such a large database can be troublesome. This searching is more difficult for beginning learners, who sometimes find it difficult to describe their needs.

1.2 Educational content linking

Therefore in this thesis we propose a third way: educational content linking. In this framework learning contents that are scattered throughout different types of course materials, such as lecture videos, slides, discussions forums, or quizzes, are linked based on their conceptual relations. A tree is then built based on these links and presented to learners along with the content. Visualizations of these relations guide learners through the content. It is hoped that this will help learners to find appropriate content for their various learning needs with less effort, and thus the learning path will be tailored to suit their background. Furthermore, this framework has two extra upsides. First, since we do not limit this framework to any types of material, educational content linking works seamlessly with both approaches described in the previous section. Second, since a conceptual relation is the only property that must be inferred, the framework is simple enough to be realized with state-of-the-art machine learning and human language technologies (HLTs). The simplicity of the framework also suggests that this model will work well in general cases rather than certain constrained environments.

In Fig. 1-1 we illustrate how educational content linking works by comparing course materials presented in the traditional way to those in the proposed "linked" way. In the figure, different types of materials are represented in different colors. Content in each type of material is segmented into smaller units, called learning objects in this thesis, and represented as nodes here. In this framework, the only requirement for learning objects is that an object should convey concepts in a self-contained way
that learners can understand. Thus, in implementations of the framework an object can be any reasonable unit such as a textbook section, a discussion thread, or a video vignette.

The left-hand side of Fig. 1-1 illustrates how materials are presented in MOOCs conventionally. Objects are aligned in sequence based on a syllabus, table of contents, or user-created time. Various types of materials are made available to learners as disjoint entities. In this scenario, a student interested in a specific concept cannot easily look up relevant information from various materials, e.g., from lectures or slides to sections of the textbook or discussions. In addition, the user-generated content, such as discussions, is usually too voluminous to be accessed efficiently if only organized chronologically.

In contrast, in educational content linking, the courseware is linked across material types and presented as a tree, as illustrated on the right-hand side of the figure. In this tree, one type of course material is specified as the trunk, here denoted by red nodes. This type of material is utilized to extract the syllabus represented by the trunk. The rest of the materials are employed to build the leaves of the tree. Each leaf, denoted by the blue and green nodes, corresponds to a learning object that is related to an object from the trunk material. In this framework, conceptual relations among learning objects are visualized in addition to the original sequential presentation of materials. Thus, we expect learners to be able to better compare content from varied
The goal of this thesis is therefore to prove our hypothesis: educational content linking helps learners find desired information at scale. We focus our investigation on two research questions: 1) Would it help learners if we are able to link course materials using human annotators? and 2) Can the courseware be linked at scale using machine learning methods? Figs. 1-2 and 1-3 outline the steps we take in this thesis to approach these two questions.

Fig. 1-2 shows how we proceed with question 1. In the investigation, we first choose MOOC subjects to focus on and collect the corresponding course materials. Human annotators are then recruited to label conceptual relations among learning materials, thereby more efficiently identifying information that is useful for them.
objects in these materials. After that, we design an interface to present to users the material content along with the linking. Using the interface, user studies are conducted to observe how learners accomplish learning tasks with different strategies for material presentation. We recruit workers from the Amazon Mechanical Turk (Turkers) [90] to carry out the study on a massive scale at a reasonable cost. Task results are analyzed to explore how linking affects learning.

As for the second question, our investigation is summarized in Fig. 1-3. We adopt a similar pipeline in approaching this question, except that we replace human annotators with a machine learning algorithm to label the linking. This automation makes the implementation of educational content linking a scalable procedure.

1.3 Contributions

The primary contributions of this thesis can be summarized as follows:

- We propose a framework for courseware presentation that allows learners to navigate much more easily. We have found with the proposed approach that learners, especially novices, can find the desired information faster without sacrificing accuracy, and can retain concepts more readily. This framework can also be easily integrated into different pedagogies to further improve learning.

- We describe the development of an end-to-end study with Turkers to explore the effects of the proposed framework on learning. The pipeline is a practical solution for the investigation of various pedagogies on a massive scale.

- We propose a method based on machine learning and human language technologies, or HLTs, to discover linking automatically. We show that this makes scalable the implementation of educational content linking, at least for STEM (science, technology, engineering and math) courses. Results suggest that learners benefit even from linking that are labeled automatically, albeit with a slightly smaller improvement than with the handcrafted system.
1.4 Thesis overview

The remainder of this thesis is organized as follows:

- Chapter 2 lays the groundwork for educational content linking by covering the related research in education and HLT. It also provides descriptions about the MOOCs and course materials used in this thesis.

- Chapter 3 describes in detail how we approach the first research question: Can linking help learning? We discuss the annotation of linking, the implementation of an interface which presents the course content and conceptual relations, the conducting of the user study, and the results.

- Chapter 4 presents an automatic linking method based on machine learning and HLTs. By analyzing how linking labeled with this method affect learning, this chapter investigates the second research question: Can linking be done at scale?

- Chapter 5 reviews the experiments and contributions of this thesis, and proposes directions for future research.
Chapter 2

Background

This chapter gives background concepts of the three main building blocks in this thesis: **learning science** to motivate the entire framework as well as supervise the system design and learning interface implementation; **crowdsourcing and learnersourcing** to recruit participants and effort at scale; **machine learning and understanding** to fuel the automation of the system. We review related literature and offer background material in these three domains. Additionally, a description of course materials used in experiments throughout this thesis will also be provided.

2.1 Learning science

For many years education practitioners and researchers endeavor to discover better ways of learning from a variety of aspects [46, 5, 109]. Researchers try to unveil the mechanism of learning, knowledge acquisition, and long-term memory establishment from cognitive science and psychology; practitioners design theory-grounded and evidence-based approaches in their classes to improve student performance. The mental state of learners and its effect on learning is one of the most discussed topics. Constructive struggle shows positive impact on learning performance by keeping learners in a mental state of boredom and confusion alternatively [35, 115, 110]. Jean Piaget proposed a theory describing how cognitive disequilibrium, such as confusion, can drive a human to develop new knowledge schema or rebuild an existing one,
i.e., motivate the process of learning [62]. However, without being properly resolved in time, confusion can lead to frustration or even dropping out [110, 136]. Active learning emphasizes the engaging of learners in discussion, high-order thinking, problem solving, or peer teaching [13]; it has demonstrated a positive impact on learning outcomes by increasing enthusiasm in students and maintaining their interest in the course [103, 111]. Cognitive load theory suggests that a complicated learning environment can overwhelm limited working memory of a human, cause distraction or frustration, and is detrimental to learning outcomes [74, 60, 124]. Although these theories seem contradictory at first glance, all of them imply the importance of balancing between challenging learners with confusion and easing their load with a proper response.

2.1.1 Tutoring at scale

Due to the delicacy of the learning mechanism, one-to-one tutoring, which is the model where learners can receive maximum attention from teachers, has set a benchmark in education that is hard to match [12]. However, a one-to-one model is cost prohibitive. In order to provide quality education to each and every learner, the idea of intelligent tutoring systems (ITSs) has been proposed [129, 102, 4]. ITS is a computer system that provides immediate feedback or hints to students based on their current learning states. For example, when asked to write a piece of code solving "square root of a number x" with guess-and-check algorithm, in an ITS students can first choose strategy "start with a guess, g". After implementing corresponding code, students can choose following strategy such as "check whether g times g equals x", "claim g as the answer", or "make a new guess". The system gives feedback such as congratulating learners, asking to try again, or providing hints either on each step or waiting until the students have submitted solutions. The example here shows an ITS applied in a problem-solving task. Actually, the framework can be implemented for different tasks to assist students in different learning stages [16, 15]. Since the tutoring is based on a computer, ITS can help many more learners at the same time.

Although ITS has been shown to be effective on improving learning outcomes
at much larger scale than one-to-one tutoring [129], authoring such a system takes a lot of effort [102]. A great tutor is made of an abundance of knowledge derived from years of his/her experience in teaching. Thus, codifying the knowledge and designing instruction strategies in the system, e.g., deciding what exactly the feedback to the learner is, is an extremely complex task. While solutions involving automated methods such as machine learning exist for some components in the system [22, 44, 31], the state-of-the-art artificial intelligence techniques are insufficient in solving the entire problem. Thus, it is usually done by handcrafted rules to codify instruction strategies. Because of the effort that has to be taken, authoring an ITS from scratch is still expensive.

Because of the demand for human input, a peer-to-peer model is proposed for scalability. Learnersourcing demonstrates how learners can collectively contribute to improving learning material and interfaces for future learners, and engage in a meaningful learning activity simultaneously [63]. Mitros and Sun presented a similar framework that allows a community of students and instructors to jointly create and polish tutoring resources around a shared skeleton [94]. Glassman and others demonstrated that learners can work collaboratively, generating rich problem solving hints and strategies [41], as well as designing complex assessment questions [93]. By automatically ranking submissions of a coding problem based on stylistic mastery from novice to experts, AutoStyle can provide students the "just a little better" submissions from others to improve their coding style incrementally [24]. The model of peer grading is another frequently applied strategy to offer learners feedback at scale with minimal instructor input [127, 101]. In addition to receiving knowledge and feedback passively, learners can also take the initiative and seek help from communities in a course forum [71] or even a question-answering (Q&A) site such as Stack Overflow [119]. In this peer-to-peer model, tutoring resources are created by employing the wisdom from a massive learner body, and thus the required efforts from instructors or experts are greatly reduced. Furthermore, the opportunity of reflecting on others’ confusions and preventing the curse of knowledge are the other two pluses [41]. The former allows learners to revisit and rethink their understanding, and the
latter bridges the gap between learners and instructors, who sometimes cannot put themselves in students’ shoes [18].

However, content created with this model usually suffers from information overload and chaos [135]. For example, a discussion board in a MOOC may have thousands of users and thousands of simultaneous threads, with great response time and quality. But for a learner who is three days behind in the course schedule, it is already impossible to follow existing discussions [53]. The peer-created materials are overwhelming and cause confusion. Inspired by the requirement of helping students receive suitable responses from an exploding amount of learning content, researchers have begun to explore a scalable means for organizing peer-generated content. Asking peers to tag content they generated is one frequently adopted strategy [9], but sometimes criticized for the lack of accuracy and consistency [108]. Wise et al. introduce an automated algorithm to identify forum posts that are related to course topics [135]. The detection of structure in discussion threads with natural language understanding is also investigated [26, 122].

This thesis proposes a framework of responding to learners’ confusion with well-organized learning content. In this framework, linking among content is discovered automatically and visualized when learners seek help. We aim to help learners resolve confusion by providing guidance for content navigation. Content generated by instructors and peers are both used, which illustrates the generalizability of our method. A user study is also explored to provide evidence of benefit in learning.

2.1.2 Course navigation

In this thesis, we introduce a method of automatically organizing learning content as well as the resolution of learners’ confusion with guidance for navigating content. The importance of guided instruction in teaching is discussed in detail by Kirschner et al. from aspects of human cognitive structure and the expert-novice difference [65]. Furthermore, due to the distant nature of MOOCs and online learning, learners usually depend on self-regulated learning to resolve their own learning needs, and whether the self-regulated learning can be achieved is highly correlated to the effi-
ciency of finding desired learning materials. Hence, there is a rich thread of research in providing guidance for navigating online learning content.

Kim demonstrates how to extract structure from learning videos with learners’ collective video interaction and annotation data [63]. With the possibility of non-linear navigation of videos, which is empowered by the extracted structure, learners reported a better learning experience. The LinkedUp project aims at linking open education resources through the use of Uniform Resource Identifier (URI) and Resource Description Framework (RDF) to improve access of content [43]. In Adaptive Educational Hypermedia, materials are organized using a concept map diagram [14, 30]. Study navigator supports the simultaneous access to multiple textbook sections, one for the current concept to learn, and the rest for background knowledge [2]. The alignment between textbook and lecture videos [92], and the restructuring of encyclopedic resources [88] are also proposed for better navigation. This thesis offers an end-to-end study in content organization and navigation, from the idea of linking and the algorithmic method, to the visualization of relationships and user study.

2.2 Crowdsourcing

In the previous section we discussed the peer-to-peer model of tutoring. This model is actually an application of crowdsourcing. A typical crowdsourcing system relies on crowd workers recruited from the Web (e.g., workers on Amazon Mechanical Turk [90]) to provide human computation for complicated parts (usually the parts that cannot be easily solved with a computer) in the system. By taking advantage of the large-scaled online community, huge problems can be divided and solved at much lower costs.

Wikipedia is one of the most compelling examples of crowdsourcing. This project of recording all human knowledge in the form of an online encyclopedia solicits contributions from anyone with an Internet connection. Since its launch in 2001, its repository now accumulates over 5.2 million articles with comprehensive topic coverage [96]. Games with a purpose (GWAP) proposes the idea of embedding work
Researchers disguise a computation problem as an online game. While people play the game, they are actually serving as processors in a giant distributed system and solving the problem without consciously doing so. The ESP game is one of the earliest successes in GWAP [132]. In the ESP game, two players are shown the same picture and they have to independently label the picture with words. Players can earn scores when their labels are matched and the goal of the game is to maximize earned scores within a fixed period of time. The real computation problem behind the game is labeling images with natural language, and players generated annotations for almost 300,000 images in its first four-month period of deployment. Other examples of crowdsourcing projects include translation (e.g., MIT OpenCourseWare [http://ocw.mit.edu/courses/translated-courses/]), or talks in TED conferences [http://www.ted.com/participate/translate], helping scientific discovery (e.g., Foldit [https://fold.it/]), or public health (e.g., Food Source Information [http://fsi.colostate.edu]). These projects are driven by noble goals (such as the public good) or offering personal benefits (such as fun). These motivations attract a large number of people on the Web and make recruitment of the crowd possible.

This thesis contributes to this line of work through two crowdsourcing applications: we utilize learning content generated by peers (i.e., course forum) for confusion resolution and recruit online workers as subjects in experiments. In the former application, the incentive for learners to contribute is that their work can not only help their peers, but also themselves and future learners. As for the latter application, workers are partially motivated by the opportunity to learn from MOOC materials.

2.2.1 Micropayment workforces

However, not every project has a goal that can attract the general public to contribute, and it is usually difficult to design a win-win condition for both researchers and the crowd. A more general approach is to pay the crowd to complete tasks, and there are many online crowdsourcing platforms offering services of matching and payment between task requesters and anonymous online workers. These platforms include Ama-
zon Mechanical Turk (AMT) [90], CrowdFlower [https://www.crowdflower.com/], and InnoCentive [https://www.innocentive.com/].

These platforms are widely used among data scientists in academia and industry to access the online workers for a variety of tasks. McGraw demonstrates an organic automatic speech recognition systems trained upon spoken utterances collected from the crowd [89]. PlateMate collects object tagging and natural language description for food photos from paid online workers, and allows users to upload photos of their meals and get information about the food intake, composition, and nutrition [95]. Callison-Burch presents a crowdsourcing workflow to evaluate quality of a machine translating system [17]. There is much other research examining the usefulness of these paid crowds for collecting, annotating, enriching, and evaluating data, including collecting spoken caption of images [45], annotating intention in user-generated content [84], real-time captioning of spoken content [73], and user interface evaluation [66].

In this thesis, we utilize the paid online workers recruited on AMT as experimental subjects in two research domains: natural language data annotation and user study in education research. For the first domain we design workflows in which workers have to understand natural language content in learning objects and label relations among these objects; for the second domain, we design tasks meaningful in learning for workers to complete and measure workers’ performance. By providing monetary incentives to the crowd, we are able to complete experiments at a much faster rate.

2.2.2 Quality control

Quality is the most criticized issue of crowdsourcing. Because of the variance in workers’ expertise, level of skills, effort, and personal bias, crowdsourcing usually yields noisier results than a conventional paradigm [76]. Furthermore, the geographically disparate nature of crowdsourcing makes it more difficult to communicate the task guideline and keep workers on consistent procedural executions than in a controlled environment such as a laboratory. Hence, there is a rich thread of research in studying how to obtain satisfactory results with crowdsourcing.

According to Allahbakhsh et al., these approaches for quality control can be cat-
ategorized into two general types, design time and run time [3]. The most common design time approach is filtering workers based on their profile, such as their previous task acceptance rate, their IP address as an estimator of their first language, or performance in a qualification test. The profile filtering is supported in most crowdsourcing platforms. Another design time approach is effective task preparation. This approach investigates how to improve quality through different worker incentives as well as better task description, workflow, and interface. Mason and Watts found intrinsic incentive such as enjoyable tasks has a more positive effect on the result than extrinsic incentive such as monetary rewards [87]. Learnersourcing described above is one of the best examples to offer workers intrinsic incentives [63, 93, 41, 94]. For a better workflow, CrowdForge proposes a framework to divide complex problems into micro-tasks [67]. Since workers on crowdsourcing platforms are more familiar with simple and independent tasks, this dividing strategy has a positive impact on the results. Chen et al. also discuss in detail the importance of clear task description (e.g., the experimental goal, who is eligible, how the result will be reviewed, and the reward strategy) for the quality [21].

Run time approaches are another type of quality control strategy. The most common way to do so is that experts review the results, and decide which ones are not qualified and should be rejected. This review mechanism is supported in most crowdsourcing platforms today. Another common approach is majority consensus. By introducing redundancy and overlapping in task assignment, majority voting can be employed to decide the real results. Karger et al. introduce a probabilistic approach to model the noisy answers from workers and improve quality [61]. There also exist studies that redesign the workflow to control quality on the fly. Lee and Glass demonstrate a multi-stage speech transcription system [77]. In this system, after each stage of transcription a machine-learning-based low quality detector is trained to filter spammers and provide instantaneous feedback to workers. Many studies have reported that, with proper quality control, crowdsourcing can yield good or near expert-level task results [77, 59, 100].

This thesis adopts a wide range of quality control approaches to improve reliability
of online workers, including majority consensus, expert review, as well as clear task
description, workflow, and interface. Moreover, in addition to monetary incentives,
our tasks also provide the opportunity to learn from MOOC materials.

2.3 Machine learning and human language understanding

The crowd can solve many complex computational problems at reasonable costs.
However, it will be more efficient if we can solve one problem and apply the so-
lution to other similar problems. This is made possible by the recent progress in
machine learning [10]. Given data that records computational problems and usu-
ally the corresponding solutions provided by a human, research in machine learning
explores algorithms or models that can summarize regularities and patterns in the
data, and solve relevant but unseen problems with discovered regularities. Thus, with
machine learning we can build a model from data annotated by a human (either a
trained data scientist or naïve online workers), and apply the model to solve future
in-domain problems automatically.

2.3.1 Human language technology

Machine learning is one of the most active research fields in computer science nowa-
days, and it has extremely diverse applications: stock market prediction [23], credit
card fraud detection [19], medical diagnosis [69], and intelligent robotics [68] to name
just a few. Among these applications, human language technology (HLT) is one of
the domains receiving the most attention.

Human language is pervasive in our daily life, and it is one of the most crucial
means for communication and information exchange. Since human language is ubi-
quitous, there is a rich thread of research concerning HLT, investigating the producing
and understanding of human language as well as attempting to improve human-to-
human and human-to-machine communication. HLT is an interdisciplinary field that
includes natural language and speech processing, computational linguistics, statistics, and psychology. Due to the recent progress in machine learning, there is a trend of applying machine learning techniques to solve HLT problems. For example, Liu investigated machine-learning-based approaches to facilitate the access of rich user-generated content online [82]. Shahaf et al. propose an algorithm to glue pieces of information scattered in various news articles, and create a structured summary for the entire story [116]. Other applications of machine learning in HLT include information retrieval [85], automatic speech recognition (ASR) [138], semantic tagging [83], topic modeling [33], and automatic question answering [131].

Since human language is an integral part of education for knowledge transferring, there is also research studying how to improve communication between learners and instructors by understanding the natural language content in learning materials with the aid of machine learning. Glass et al. demonstrated the MIT Lecture Browser. By automatically transcribing speech in lecture videos with ASR techniques, learners can easily browse through the text and identify topics they are interested in more efficiently [40]. On top of the transcribed text, in the FAU Video Lecture Browser, key phrases are also extracted, ranked, and presented along with aligned lecture video. By clicking each key phrase, learners can access corresponding video vignettes for detailed discussion [105]. Without transcribing speech to text beforehand, a method matching spoken search queries to lecture speech directly on audio is also proposed to improve video navigation [107]. Fujii et al. further presented an algorithm to automatically summarize course lectures; thus learners can get the big picture behind each lecture without watching the video from beginning to end [37].

Beyond the lectures, there also exist studies in understanding of textbook and course forums with HLT and machine learning, since there is an abundance of natural language in these materials. Lin et al. proposed a method to classify genres of discussion threads for improving accessibility of forums [81]. A similar idea is applied to identify questions and potential answers in discussion boards [54]. Li et al. demonstrated how to build a semantic forum that allows semantic search, relational navigation, and recommendation with HLT [80]. An automatic approach to discover
relevance among textbook sections was also investigated [2]. Due to the popularity of MOOCs, recent research has begun using HLT in understanding MOOC materials, including intention classification and topic modeling for forum posts [122, 135, 121, 39], textbook section recommendation for lecture videos [92], and automated essay scoring [7].

These research strategies demonstrate that, with HLT, the machine can learn to understand course materials as well as assist the information exchange among teachers, students, and materials. Due to these advantages and the nature of MOOCs (i.e., size of the audience and physical distance among them), HLT can play a crucial role in improving the learning experience and performance in online learning. In this thesis we introduce an HLT-based method to automatically discover relations among various types of MOOC materials, and show its benefit in learning.

2.3.2 Conditional random fields (CRF)

In this thesis we adopt conditional random fields (CRF) to model the relations among learning objects. CRF is an instance of graphical models [123], which is a graph designed to model the conditional dependence structure among random variables (a random variable is usually used to express the observation in data samples and the hidden classes these samples belong to). The training and inference of CRF is well studied in the machine learning field. Therefore, it is widely used in learning temporal dependence from sequential data, such as speech, text, image, and bioinformatics [83, 47, 113]. Since most course materials can be expressed with sequential structure, we believe the CRF is a perfect match to our problem. In the following we will introduce the mathematical definition, the training, and the inference of CRF.

A general CRF can be defined as follows: given $Y$ as the set of unobserved variables, and $X$ as the set of observed ones, let $G$ be a factor graph over $X$ and $Y$. If, for any $x$, the conditional probability $p(y|x)$ can be factorized according to $G$, then
(X, Y) is a conditional random field [123]. That is,

\[ p(y|x) = \frac{1}{Z(x)} \prod_{a=1}^{\Lambda} \Psi_a(y_a, x_a) \]  

(2.1)

where \( y \) is a vector denoting the assignment to \( Y \), \( x \) denoting the assignment to \( X \), \( \Psi_a \) is the set of factors in \( G \), \( a \) is the index of factors, and \( Z(x) \) is the normalization term.

\[ Z(x) = \sum_{y} \prod_{a=1}^{\Lambda} \Psi_a(y_a, x_a). \]  

(2.2)

Each factor \( \Psi_a \) is a function of \( y_a \) and \( x_a \), which are subsets of the unobserved and observed variables respectively (i.e., \( y_a \subseteq Y \) and \( x_a \subseteq X \)). The value of \( \Psi_a \) is a non-negative scalar, which can be interpreted as a measure of how compatible this subset of assignment \( y_a \) to the unobserved variables is with its dependent observations \( x_a \). An example of a general CRF and its corresponding factor graph is shown on the left panel of Fig. 2-1. In this graph, \( \Psi_1 \) depends on \( X_1 \) and \( Y_1 \), and \( \Psi_2 \) depends on \( Y_2, X_1, \) and \( X_2 \) for instance. Since there is no constraint to the underlying factor graph of CRF, we can see it is flexible in expressing various structures among data.

With equation 2.1, inferring labels (i.e., unobserved variables) from observations can be modeled with a maximization problem: finding the label assignment \( y \) which maximizes the conditional probability \( p(y|x) \) given the observations \( x \). However, solving this maximization problem in general CRFs is intractable [123]. There are...
two usually adopted approaches to obtaining feasible solution. First, if we limit the underlying factor graphs of CRFs to several special cases, e.g., a chain or a tree, the exact inference can be solved in polynomial time. On the other hand, several algorithms can be used to obtain approximate inferences, e.g., Markov chain Monte Carlo sampler [47], and loopy belief propagation [125]. Since a linear sequence is the most common and dominant structure for arranging topics in course materials, we choose the linear chain CRF in this thesis and design our algorithm based on it. Another benefit of a linear chain architecture is that it reduces the model complexity and mitigates overfitting. This is crucial, especially when annotated training data is hard to obtain, such as in our problems.

In the right panel of Fig. 2-1, we show an example of linear chain CRFs. Similar to general CRFs, $X$ and $Y$ also represent the observed and unobserved variables respectively, except that $Y_t$ are structured to form a chain. This chain structure adds a constraint to the probabilistic dependence expressed by the model that an unobserved variable $Y_t$ can directly depend on only the single previous unobserved variable $Y_{t-1}$ and several observations $x_t = \{X_{ts}\}_{s=1}^{N(Y_t)}$. Here $N(Y_t)$ denotes the number of observed variables depending on $Y_t$.

With the linear chain structure, the conditional probability $p(y|x)$ can be rewritten as following equation

$$p(y|x) = \frac{1}{Z(x)} \prod_{t=1}^{T} \Psi_t(Y_t, Y_{t-1}, x)$$

by replacing $\Psi_a$, the set of factors in $G$, with $\Psi_t$. Each factor $\Psi_t$ is a function of $Y_t, Y_{t-1}$ and $x$, and these factors represent the linear-chain factor graph. In real application $\Psi_t$ is usually set as the following form $\Psi_t(Y_t, Y_{t-1}, x) = \exp\{\sum_{k=1}^{K} \theta_k f_k(Y_t, Y_{t-1}, x)\}$, and Equation 2.3 is rewritten as

$$p(y|x) = \frac{1}{Z(x)} \prod_{t=1}^{T} \exp\{\sum_{k=1}^{K} \theta_k f_k(Y_t, Y_{t-1}, x)\}.$$  

Here $f_k(Y_t, Y_{t-1}, x)$ is a feature function that researchers need to design based on domain knowledge, and $\theta = \{\theta_k\}_{k=1}^{K}$ is the parameter set that has to be learned from
training data. This chain structure is called the Markov property, which assumes the modeled stochastic process is memoryless, i.e., the prediction to the current unobserved variable depends only on the prediction to the previous one, and no other earlier prediction. Another popular model that assumes this property to hold is the hidden Markov models (HMM), and in fact this linear chain CRF can be interpreted as a generalized HMM, where the factor function $\Psi$ does not need to have a probabilistic interpretation as HMM does. With the memoryless property, the inference problem of linear chain CRF (as well as HMM) can be solved efficiently with a dynamic-programming algorithm [123].

In addition to the inference problem, another issue of applying CRFs to real tasks is parameter estimation, or training. The maximum likelihood criterion is typically used for estimating parameters: given the fully labeled training data $C = \{x^{(i)}, y^{(i)}\}_{i=1}^N$, where $(x^{(i)}, y^{(i)})$ is the $i$-th sample in the data, $x^{(i)} = (x_t^{(i)})_{t=1}^L$ is a sequence of observations, and $y^{(i)} = (Y_t^{(i)})_{t=1}^T$ is a sequence of labels corresponding to $x^{(i)}$, we estimate the model parameter $\theta$ with the maximum likelihood estimator $\hat{\theta} = \arg\max_{\theta} l(\theta)$. $l(\theta)$ is the objective function and equals $\sum_{i=1}^N \log p(y^{(i)}|x^{(i)}; \theta)$ with $p(y^{(i)}|x^{(i)}; \theta)$ as defined in Equation 2.4. However, in general the estimator does not have an analytic form. Therefore, a gradient ascent approach is adopted to obtain an approximate solution to this optimization problem (other approaches also exist but gradient ascent is most commonly used in practice). The algorithm for gradient ascent can be summarized as follows:

**Algorithm 1** Gradient ascent algorithm

1: Randomly initialize the parameter set $\theta$

2: repeat

3: Compute the gradient of the objective function, $\nabla l(\theta)$

4: Update the parameter set $\theta$ according to pre-defined learning rate $\rho$

$$\theta := \theta + \rho \nabla l(\theta)$$

5: until convergence criterion is achieved.

This algorithm updates the estimated parameters along the direction where the objective function is increased most at each step. When the convergence criterion (e.g., the difference of estimation in two consecutive iterations is less than the pre-
defined threshold) is achieved, the estimation is the trained model parameters. There
are many variations of this algorithm, such as Newton's method, BFGS, and conjugate
gradient [123]. These variations attempt to improve convergence speed with different
techniques but share the same compute-gradient-and-update idea.

These training and inference techniques are widely used in various applications of
linear chain CRF. In Chapter 4, we will discuss how to apply the general model in
our problem of linking discovery.

### 2.3.3 Word embedding

In order to apply statistical models to natural language content, we have to represent
content in a form that the model accepts, i.e., numeric vectors. Here we give an
introduction to the vector representations employed in this thesis.

The first representation is unigram embedding, or Bag of Word (BoW) embedding.
In this simple embedding, a document is represented as a vector \([N(w_1), N(w_2), ...,\)
\([N(w_{|V|})]]\), where \(w_i\) is the \(i\)-th word in vocabulary \(V\), and \(N(w_i)\) is the score of \(w_i\) in
this document. The score can be the number of occurrences, the word frequency, or
term frequency-inverse document frequency (TF-IDF) [112]. The upside of unigram
embedding is that this method is intuitive and easy to train. However, since each
word is represented as an atomic unit in the vector and different words are encoded
independently, the long-range lexical dependency, such as the context of a word, is
missing in this representation.

To make the shallow and local representation embed lexical dependency in a longer
range, we can adopt an \(n\)-gram model, which is an extension of unigram embedding,
and each element in the vector represents a combination of \(n\) words instead of a
single word. Nonetheless, this model provides only limited added value. An \(n\)-gram
model greatly increases the dimension of vector representation, since it exhaustively
enumerates all possible combinations of \(n\) words. Due to the curse of dimensionality,
in practice we can only use a small \(n\) in order not to overfit, especially when the size
of training data is limited. Thus, the range of dependency this method can encode is
still restricted.
We turn to word2vec embedding for our second representation with longer lexical dependency [91]. Word2vec is a two-layer neural network. Its input is a text corpus and its output is the vector representation for each word in the corpus. As compared to the n-gram and unigram method, word2vec is a continuous language model, which means that each word is represented as a continuous vector. The upside of this representation is its capability of encoding semantic and syntactic dependencies among words. In the n-gram or unigram method, each word or combination of words is represented with an independent element in the vector, and the relations among words cannot be encoded efficiently; in word2vec, the neural network model is designed to discover and represent semantic and syntactic dependencies from patterns from words’ context. For instance, based on the word2vec model trained on millions of Wikipedia pages, \( v_{\text{King}} - v_{\text{Man}} + v_{\text{Woman}} \approx v_{\text{Queen}} \), and \( v_{\text{Apples}} - v_{\text{Apple}} + v_{\text{Car}} \approx v_{\text{Cars}} \), where \( v_i \) denotes the word2vec representation of word \( i \). With word2vec embedding, the document representation can be simply obtained by averaging word vectors over the entire document.

The word2vec embedding is trained using a feedforward neural network model with architecture shown in Fig. 2-2 [106]. In the figure, \( x_t \) is an one-hot vector with its \( i \)-th element equal to \( \delta(w_t = v_i) \); \( W^1 \) and \( W^2 \) are matrices of weights to be learned from a corpus; \( h \) is a vector of hidden layer projection obtained by transforming the hidden layer input with the sigmoid function \( \sigma \); \( k \) is the hyper-parameter deciding the size of context for this model to learn from. Here \( \delta() \) is an indicator function with \( w_t \) as the \( t \)-th word in corpus and \( v_i \) as the \( i \)-th word in vocabulary. These vectors and parameters are related to each other based on the following equation:

\[
x_t = W^2 h = W^2(\sigma(W^1[T_{t-k}, T_{t-k+1}, ..., T_{t+k}]^T)).
\]  

(2.5)

This model can be interpreted as a classifier trained to predict a word based on its neighbors, typically using the Stochastic Gradient Descent (SGD) training algorithm. The algorithm is very similar to the one introduced in Section 2.3.2, except that, in each step, we update the estimated parameters \( W^1 \) and \( W^2 \) along the direction where
the objective function is decreased most, and we use cross entropy as the objective function. After the training, the vector representation of $w_t$ is $W^1 x_t$. In this way, the neural network can encode the long-range semantic and syntactic dependencies in vectors by discovering patterns from the context of words.

In fact, there are other common approaches to represent text documents as vectors that encode high-level lexical and semantic information: doc2vec [75] and topic modeling [98, 52, 11]. Doc2vec is a very similar algorithm to word2vec except that it learns representation for larger blocks of text directly, such as paragraphs or sentences. Topic modeling refers to a family of methods for discovering latent semantic
structure and identifying the subsets of words co-occurring more frequently in documents of various "topics". In this thesis we choose not to use these representations. Doc2vec requires too much in-domain data for training. The learned "topics" in topic modeling are too broad for our problem. For instance, with topic modeling we can easily identify *python*, *complexity*, or *object oriented programming* corresponding to topic *computer science*, and *standard deviation*, or *hypothesis testing* belonging to *statistics*. However, when it comes to distinguishing *complexity* from *programming*, since the two concepts are in different lectures, topic modeling usually introduces a lot of noise. Hence, we surmise that doc2vec and topic modeling are not suitable for our problem.

### 2.4 Corpora

Before starting to implement our proposed framework and investigating the effects on learning, we have to first decide which materials and MOOC subjects our system should be built upon. Today there are over four thousands MOOCs on the Web covering subjects from science to humanities. Types of materials and pedagogies adopted in these MOOCs are diverse. It is impractical to expect an exhaustive exploration of every condition. Thus, in this thesis we make a tradeoff between feasibility of experiments and generalizability of results. In the following we discuss the decisions we make and the rationale.

#### 2.4.1 Course subjects

Experiments in this thesis use two MOOCs: Introduction to Statistics: Descriptive Statistics (Stat2.1x), and Introduction to Computer Science and Programming (6.00x). Stat2.1x was offered by University of California, Berkeley, from February to March in 2013 on edX [55], and 6.00x was offered by Massachusetts Institute of Technology (MIT), from February to June in 2013 on edX [1]. Stat2.1x is an introduction to fundamental concepts and methods of statistics, which require basic high-school level Mathematics. 6.00x is aimed at undergraduate students with little programming
experience, and discusses how to solve real problems with computational approaches and computer programming. Both MOOCs were very successful. Stat2.1x has over 47,000 registrants, and 6.00x has over 72,000 registrants. Due to the popularity of these two MOOCs and the growing interest in STEM education these years, we choose to focus investigation in this thesis on Stat2.1x and 6.00x. The popularity of underlying MOOC subjects makes findings in experiments more influential, representative and likely to be applied to different conditions. Furthermore, our familiarity with the topics is another plus.

In our following investigation, these two MOOCs serve different purposes. We use Stat2.1x for developing minimum viable product [58] and 6.00x for the final evaluation. In system development, minimum viable product is an intermediate stage where a product with a minimum amount of features is built to gather information and user feedback about the product. In this stage, the goal is to validate product ideas from interaction with real users with minimum cost. This provides insights for further system development and greatly reduces risk as compared to implementing all features in the product at once. We believe Stat2.1x can serve this purpose well for two reasons. First, this MOOC is shorter (less than two months) than most of the others, but still contains necessary components and course materials. Therefore we can implement our framework on a complete MOOC more readily, e.g., labeling linking on fewer materials. Second, statistics is familiar and interesting to many, thus making it easier to recruit experimental subjects in our study. For these reasons, we select Stat2.1x for an intermediate validation of the benefit and scalability of educational content linking. The role of Stat2.1x can be interpreted as a development set in a machine learning system.

With the validation and feedback, we improve our implementation on 6.00x and evaluate the resulting system in depth. In addition to aiming at answering fundamental questions such as whether linking is beneficial or scalable, we also explore advanced features, such as reproducibility, generalizability, and portability of the framework. We can interpret 6.00x as a test set in machine learning system. Using one MOOC for development and a different MOOC for testing makes the evaluation more credible.
and less subject to the criticism that we overfit our implementation to a particular MOOC.

There are other benefits of using 6.00x. Since this MOOC was offered by MIT, there are many more resources available to us. We can easily reach course staff and MIT students who have taken the corresponding course on campus for insightful understanding. Moreover, this MOOC and its corresponding residential course have been offered many times on edX and at MIT. These multiple offerings leave room for expanding our survey along various dimensions in the future.

### 2.4.2 Course materials

Within a MOOC, a wide range of course materials are available to learners, such as lecture videos, lecture slides, labs, textbook, discussion forum, course Wiki, quizzes and exams. Considering the development cost, again we choose a subset from these materials for experiments in this thesis: lecture videos, slides, and textbook for Stat2.1x, and the previous three materials together with discussion forum for 6.00x. There are several reasons for us to make this choice. First, these types of materials are common to many MOOCs nowadays and contain a large portion of learning content. Second, these materials have similar form over different course subjects. This fact makes the experiment easier to reproduce from MOOC to MOOC. In contrast, for example, quizzes and exams have diverse styles, from multiple-choice questions to computational problems to essay writing, and each course subject emphasizes on various styles. Thus it might be challenging to generalize experimental results to a variety of MOOCs. Third, these materials allow us to investigate various types of linking, from linking two types of materials composed by the same instructor that can be aligned in order properly, to linking two materials with very different creators and organization. One example of the first type is linking between lecture videos and slides, and examples for the second one are linking lecture videos to textbook or to discussion forum. We will explain these two types in detail in the next section. For these reasons, we believe our choice of materials can help us obtain generalizable and reproducible experimental results with reasonable cost.
Table 2.1: Summarization of sizes of course materials used in this thesis.

<table>
<thead>
<tr>
<th></th>
<th>Sizes</th>
<th>Words</th>
<th>Vocabulary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stat2.1x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lecture video</td>
<td>7 hours</td>
<td>62k</td>
<td>1,743</td>
</tr>
<tr>
<td>Lecture slides</td>
<td>157 pages</td>
<td>11k</td>
<td>785</td>
</tr>
<tr>
<td>Textbook [120]</td>
<td>77 sections</td>
<td>45k</td>
<td>1,825</td>
</tr>
<tr>
<td>6.00x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lecture video</td>
<td>21 hours</td>
<td>174k</td>
<td>3,086</td>
</tr>
<tr>
<td>Lecture slides</td>
<td>498 pages</td>
<td>32k</td>
<td>1,952</td>
</tr>
<tr>
<td>Textbook [42]</td>
<td>144 sections</td>
<td>119k</td>
<td>4,594</td>
</tr>
<tr>
<td>Discussion forum</td>
<td>1,239 threads</td>
<td>236k</td>
<td>6,772</td>
</tr>
</tbody>
</table>

Note that discussion forum is only chosen in the evaluation MOOC (i.e., 6.00x). This is because accessing data with personally identifiable information, such as forum posts, requires lengthy paperwork. This work should not be a part of development of minimum viable product.

In Table 2.1 we summarize the quantity of these materials. The first column lists the number of video hours, slide pages, textbook sections, and discussion threads. Here we measure the size of the textbook by number of sections rather than pages, since the textbook used in Stat2.1x is a Web-based electronic book, and pages in this book are not properly defined. Furthermore, considering the cost of data annotation, we only used the threads posted under the lecture videos in our experiment; these 1,239 threads are about one tenth of the total posts in 6.00x. The second and third columns show the number of words in the available material and the count of unique words, respectively. Here video transcription is used for computing the number of words.

From this table we observe that the amount of content in 6.00x is much greater than the amount in Stat2.1x. This is another reason why we chose Stat2.1x to develop the minimum viable product. The much smaller corpus means a faster process of establishing linking among the course materials.
Chapter 3

Would linking help learning?

This chapter investigates the first research question: Would it help learners if we were able to link course materials using human assistance? In MOOCs most instruction and knowledge acquisition happen as the learning content is delivered. Thus, in the previous chapter, we surmised that making materials more accessible by linking them would enhance the learning experience and outcomes. For example, when learners are confused at a specific point of the lecture, more accessible materials allow them to find useful content with which to more easily clear up their confusion. In this chapter, we explore the research question in support of our theory with empirical evidence. To approach this question, we conducted an end-to-end study investigating the following issues:

- How to link course content with human assistance?
- How to present linking along with content to learners?
- How to measure the effect of linking on learning?
- Is linking helpful?

The study is conducted on two MOOCs: Stat2.1x and 6.00x, which are described in detail in Section 2.4. In the first MOOC we evaluate the idea with minimum input, and in the second MOOC we measure system performance in realistic conditions. In
our experiment, we discover that, using human annotation, we can build an interface presenting course material with meaningful interconnection. Our interface is shown to be beneficial in supporting learners to complete their tasks: it allows learners to search for information more efficiently, retain more concepts using the same amount of time, and focus on informative learning content. Moreover, we contribute to the research community by providing a user study pipeline that can be conducted at scale and in a cost-effective way.

3.1 Linking materials

Linking is an abstract and general idea; however to implement a real system based on the idea, a concrete definition is required. Linking refers to relations among objects and can typically be visualized as a graph diagram, with vertices representing the objects and edges for the linking. However, many people are not comfortable interacting with a general graph diagram [48], since the volume of possible paths in the graph is confusing and overloads the human cognitive system [65]; in order not to distract learners, most learning content, e.g., lectures, or textbook sections, is aligned in sequence. Likewise, we limit our linking to a specific trunk-and-leaves architecture. In this section, we discuss how to link course content with human assistance under this architecture.

3.1.1 The linking tree

Fig. 3-1 illustrates the trunk-and-leaves architecture we limit linking to, with blue nodes representing the trunk and the others for leaves. A node in the diagram represents a learning object, which is defined in Section 3.1.2. The trunk visualizes the main flow of the courses and shows students a clear learning path to follow. Each leaf node attaches to one object on the trunk, and represents a supplementary learning object for the corresponding trunk node.

In this thesis, we select lecture video sequences as the trunk of the tree, since most online or residential classes center around the lecture or lecture videos. Below,
we discuss how to obtain supplementary objects, i.e., the leaves, for each trunk node with human annotation.

### 3.1.2 Homologous and heterologous linking

We identify supplementary objects for each node on the trunk by discovering the relations between three pairs of course materials: lecture videos and slides, videos and textbook, and videos and discussions. In this thesis, instead of treating the entire video as an atomic element, we discover relations at the level of the video segment. We surmise that this finer granularity is helpful in visualizing in-video structure, such as subgoals, subtopics, or meaningful conceptual pieces; the structure improves learning and navigation by summarizing and abstracting low-level details as well as reducing the cognitive load of learners [63]. In order to achieve this level of granularity, we define a learning object as a segment of the lecture video, a page in the lecture slides, a textbook section, or a discussion thread.

Before describing how to discover relations between materials, we first discuss the two types of relations: homologous and heterologous linking. The reason for discussing these two types first is because their linking patterns are distinct; the difference can greatly affect how to discover relations. In Fig. 3-2, we show examples
of these two types of linking, with homologous linking in the upper panel and heterologous in the lower panel. The figure illustrates two sequences of learning objects from two types of materials along with the relation between each object pair. Here objects in the trunk (i.e., segments of lecture videos) are represented as blue nodes, and objects from another material (e.g., pages of slides, textbook sections or discussion threads) are in orange or pink. The indices of objects are also labeled (1 to 7, A to C, and a to c).

As shown in the figure, a homologous linking is a many-to-one and monotonic (or order-preserving) mapping between two sequences of learning objects. A monotonic mapping satisfies the following attribute:

\[ x \preceq_\alpha y \text{ implies } f(x) \preceq_\beta f(y) \text{ if } f(x), f(y) \text{ is not } \emptyset \]  

(3.1)

where \( x \) and \( y \) are learning objects in material \( \alpha \), \( f(x) \) and \( f(y) \) are the objects in material \( \beta \) and linked to \( x \) and \( y \) respectively, and \( \emptyset \) is the empty set. \( x \preceq_\alpha y \) refers to the case in which object \( x \) precedes object \( y \) in the material sequence \( \alpha \); the precedence can be defined by the chapter/section/lecture indices or the thread post time. Homologous linking mostly exists between two materials authored by the same person, e.g., between video segments and slides, since in this case topic arrangement
in different materials usually follows the same ontology.

In contrast, heterologous linking refers to the case in which mappings between two object sequences are many-to-many or not order-preserving, as in the example shown in the lower panel of Fig. 3-2. In this example, the precedence of objects in one sequence is not preserved after mapping these objects to objects in another sequence, resulting in the many crisscrosses when visualizing the mapping. Heterologous linking usually exists when the underlying two materials come from different authors. This is because the cognitive system in which humans interpret and store knowledge varies from person to person. It is very likely that different authors arrange topics and content in different ways.

The linking between lecture videos and forum discussions, or lecture videos and textbook, can usually be classified as representing a heterologous relationship. For a textbook, its arrangement of chapters and sections can be totally different from the arrangement of lectures in a course. As for posts in a forum, if we sort them by creation time, they can also be in a distinct order from the lectures. This is because every learner has a different pace and experiences different learning progress; hence even at the same point in time, different learners may commence discussions about different topics.

In fact, instead of a dichotomy, it is more precise to interpret the monotonic property as a spectrum, where the proportion of mapping that violates Equation 3.1 changes gradually from zero to one. For example, although neither video-to-discussion or video-to-textbook linking are order-preserving, there are usually more crisscrosses in the former. We choose to simplify the spectrum to two conditions – homologous and heterologous linking – because it is not practical to investigate every point on the spectrum. Since the proportion of order-preserving mappings between materials is highly correlated to the complexity of identifying supplementary objects, we used two methods to discover the relations for the two types respectively.
3.1.3 Linking representation

Since homologous linking is a many-to-one and monotonic mapping, we formulate relation discovery as an alignment problem. In Fig. 3-3 we show how to annotate homologous linking based on this formulation to represent the relation configuration between materials illustrated in the left panel of Fig. 3-2. Given two sequences of materials – one the trunk and the other a set of candidates of the leaves – we identify the non-overlapping, sequential chunk of trunk nodes corresponding to each leaf in order, and label these nodes with the index of the leaf. Here, we define a correspondence between a chunk of trunk nodes and a leaf when they contain identical discussion about a concept. In addition, since the video transcription sentence is the only unit that can be obtained easily and is of a finer granularity than the entire video, we choose one sentence as a video segment (i.e., a node on the trunk).

For heterologous linking we can adopt the same formulation, as shown in the upper panel of Fig. 3-4. However, in this case, since the aligned chunk of trunk nodes need not be sequential, and the chunks for different leaves may overlap, identifying the chunks is much more complicated than in the homologous case. Furthermore, the possibility of one video segment aligning to multiple leaves makes this problem a multi-label classification problem, which increases the complexity of any automated method to infer the relation. Consequently, we propose another formulation for heterologous linking, as shown in the lower panel of Fig. 3-4.

In this alternative formulation, we divide the entire relation identification problem into several sub-problems by considering each leaf independently. That is, in each sub-problem, our goal is to discover the relation between the sequences of trunk nodes and a separate leaf. Each sub-problem is interpreted as a binary classification task,
where every trunk node is classified as related (denoted as "Y" in the figure) or unrelated ("N") to the leaf. In this way, we solve the entire problem by solving many simpler sub-problems.

We can also adopt a transcription sentence as a node on the trunk. However, the workload of relation discovery can be too heavy for the humans because we have many sub-problems to solve. Since in the two MOOCs investigated in this thesis we have both homologous and heterologous linking in the corpus, in our implementation we first annotate the former linking, and merge the sequential chunk of sentences that are aligned to the same leaf as a new video segment (for clarity, below we refer to this video unit used in heterologous linking as a "video vignette", and use "video segment" for a general purpose, e.g., a sentence in homologous linking or a vignette in heterologous). These vignettes inferred from the alignment are used as trunk nodes in the following heterologous linking to reduce annotator workloads. In addition, we define a trunk node as related to the leaf if the concept contained in the trunk node is equivalent to, an instance of, or a part of the leaf. Here, we choose a more relaxed definition than the "correspondence" of homologous linking, because in the heterologous case, content is usually organized in various manners and we are less
likely to find identical mappings in the two underlying materials.

Thus in this thesis we link MOOC materials using the following steps:

1. Determine homologous linking of video transcription sentences (trunk nodes) against lecture slides (leaves). The homologous linking is formulated as an alignment problem.

2. Group transcription sentences linked to the same slide together, and define each group as a "video vignette".

3. Determine heterologous linking of video vignettes against textbook sections and discussion forum posts. The heterologous linking is formulated as a binary classification problem.

3.1.4 Annotation tasks

For these two types of linking, we designed two websites to collect human annotations. In Fig. 3-5, a screenshot of the website used for homologous linking is shown. Since in this thesis, the only homologous linking investigated is the alignment between the lecture video transcription and slides, we thus design the interface to present for each instant a transcription of a lecture video and a deck of slides from the same lecture in parallel. In the website, a human annotator first selects a slide page by clicking "<" (previous page), or ">") (next page). Then the annotator clicks and drags on the sentences he or she intends to align to the selected slide, and clicks on the "Add the selected chunk" button to confirm the alignment. After the confirmation, sentences aligned to different pages of slides are highlighted with different background colors, which are listed on the rightmost side of the screen. For instance, in this figure the first three sentences are aligned to the first slide, and the following ten sentences are aligned to the second. The interface also provides a "Clear your alignment" button for annotators to clear the confirmed alignment. Note that in this interface we do not show the lecture video, because we intend to simplify the workflow of this annotation task, and encourage our annotators focus on the transcription sentences.
Objects

- At heart, programs will manipulate data objects
- Each object has a type that defines the kinds of things programs can do to it
- Objects are:
  - Scalar (i.e., cannot be subdivided), or
  - Non-scalar (i.e., have internal structure that can be accessed)

Each object has a type that defines the kinds of things programs can do to it.

- Scalar (i.e., cannot be subdivided), or
- Non-scalar (i.e., have internal structure that can be accessed)

Figure 3-5: Website built to collect homologous linking (i.e., the alignment between lecture video transcription and slides)

We designed another website for the annotation of heterologous linking; as an example we show its screenshot in Fig. 3-6. During implementation we investigated two mappings, i.e., lecture video to discussion forum and video to textbook, for the heterologous condition; therefore, we also designed the interface to present video content and discussions (or textbook) side-by-side. As shown in the figure, in the upper half of the website, lecture video from the entire course is presented. Annotators can access the videos by clicking on the main title (this is a title shared among several lecture videos, and is here presented in blue text) and subtitle (a title specific to each video, presented in black text) of each video listed on the left hand side of the screen. In addition to a lecture video, here we also provide the aligned video transcription on the right and the thumbnails of aligned slides below the video. The transcription is synchronized with the video based on the time code extracted from the video subtitle file. The alignment between a video and its slides is inferred from human annotation in the homologous task as well as from the time code of transcription sentences; we show this alignment information using black markers on the video scrubber, each marker of which represents the beginning of a video vignette that is aligned to one slide. As compared to the website for homologous linking, here we present video along with the various relevant materials (e.g., the aligned slides and transcription)
Figure 3-6: Website built to collect heterologous linking (i.e., the binary classification task of deciding whether a video vignette/discussion thread pair or a video vignette/textbook section pair is relevant)

simultaneously, to provide annotators with a comprehensive understanding of each video vignette, which in the heterologous task is the unit we work on.

In the lower half of the website, a discussion thread (or a textbook section) is shown to annotators. The annotator selects the relevant video vignettes from the entire course, and links these vignettes to the thread by clicking the "Link vignette" button. Linked vignettes are also shown on the screen with text in cyan. Since a video vignette $v_i$ can be defined by two markers (i.e., the marker representing the beginning of a video vignette that is aligned to slide $i$ and the next marker for slide
the annotator can simply select \( v_i \) by dragging the video scrubber to any place between the two markers.

We then recruited human subjects to annotate links in our corpus with the two websites. In Stat2.1x, the linking between lecture videos and slides (homologous) and the linking between videos and textbook (heterologous) were annotated; for 6.00x, in addition to the two pairs of materials above, the video-discussion linking was also labeled. For the statistics course, in order to expedite the development process, we employed for the labeling three annotators (including the author) who were graduate students or postdoctoral researchers with expertise in statistics. Each annotator spent five hours labeling the homologous task and 8 hours on the heterologous one. Majority voting was applied to the labeling results of the three annotators to obtain the final linking annotation used in this thesis.

To understand the consistency among annotators, we computed Cohen’s kappa statistic \cite{cohen1968coefficient} to measure the inter-annotator agreement. The statistic can be written as the following equation:

\[
\kappa = \frac{p_a - p_c}{1 - p_c},
\]

In this equation, \( p_a \) is probability of agreement among annotators observed in samples (i.e., in the corpus), and \( p_c \) is the theoretical probability of chance agreement. In Stat2.1x, the kappa scores are 0.867 and 0.599 for the homologous and heterologous task respectively. According to several arbitrary guidelines, these scores show almost perfect and moderate agreement among annotators in the two tasks respectively \cite{landis1977measurement, brezina1979agreement}. In addition, the lower score in the heterologous linking also reflects that the underlying task is more complicated than the homologous one.

For 6.00x, one of our goals was to establish a more realistic pipeline with its materials. Thus, instead of researchers, we chose to recruit online workers from AMT for the homologous linking, and teaching assistants in both the edX and MIT offering of 6.00x for the heterologous tasks. Online workers were employed here for homologous linking since they were an economic choice for data annotation with satisfactory quality \cite{our2016crowdsourcing}, especially for simple underlying tasks. In contrast, we chose to recruit
teaching assistants for the heterologous task, because in the designed annotation workflow, annotators must be familiar with the entire course before they can select relevant videos for each discussion (or textbook section) efficiently. This task requires annotators to spend a much longer period of time to ramp up. However, since on the crowdsourcing task-matching platform workers usually have thousands of tasks to choose from at the same time, the returning worker rate is much lower in comparison to recruiting teaching assistants for annotation. The low return rate necessitated a large portion of time to train new workers to be familiar with the content, and likewise a smaller portion of experienced workers yielding quality output.

We created a total of 945 HITs (i.e., Human Intelligence Tasks)\(^1\) on AMT to align 105 video-slide pairs, with nine workers on each pair and a reward of $0.25 for each HIT. 100 workers participated in the annotation; the mean and standard deviation of the total time spent for each worker were 35.5 minutes and 63.4 minutes respectively. Majority voting over the nine labeling results in each pair of video and slides was also used to obtain the alignment used in this thesis.

For the labeling of the 144 textbook sections to lecture videos, we recruited four teaching assistants from the 6.00x course offered at MIT; the labeling of 1,239 discussion threads was done by two teaching assistants from the MIT 6.00 course and five teaching assistants from the edX version. Each section or discussion thread was labeled by three different annotators for the majority consensus process. These annotators spent 7.5, 7.5, 5, and 3 hours on the textbook task, and 16, 14, 10, 10, 6, 6, and 2 hours labeling the forum. We paid these annotators at the rate of $45 per hour.

We also computed kappa scores for these annotations. For the video-to-slide, video-to-textbook, and video-to-discussion linking, the scores were 0.810, 0.761, and 0.434 respectively. We are satisfied with these results because all of the scores also show almost perfect or moderate agreement among annotators [72, 34]. Comparing these numbers to the scores obtained in Stat2.1x, we find that in homologous linking, online workers were able to yield as consistent annotation as the researchers (cf. 0.810

---

\(^1\)On AMT, each HIT is a self-contained task a worker completes for a reward.
vs 0.867); annotators were more consistent in linking textbook to video sequences (cf. 0.761 vs. 0.599), presumably because the author of the textbook used in 6.00x was also one of the lecture instructors. Therefore, it was easier to identify the related learning objects from two material sequences. In contrast, linking discussions was undoubtedly the most complicated task since the learner-generated content was noisier and less organized than that from educators, as reflected in its low inter-annotator consistency. Even so, the 0.434 kappa score still shows fair agreement among teaching assistants.

3.2 Presenting linking to learners

With the linking among course materials annotated, we then designed an interface to visualize the annotated relations while learners access the course content. Our ultimate objective was to provide learners guidance and make learning content more accessible, and to help them find supporting materials more efficiently when they were confused or otherwise in need. After surveying the relevant literature [63, 122, 135, 65, 92, 121, 39, 86], we identified three high-level goals that informed our design.

Supporting relational navigation among materials. Many observations suggest that, in current MOOC platforms, it is difficult for learners to identify related materials [122, 135, 92, 121, 39]. Thus many interactions are not supported, such as skipping redundant forum posts, or navigating from a specific point of a lecture to further discussion in the forum and detailed explanation in the textbook. To support these navigation needs, we leveraged the annotated linking among learning materials. We designed our interface to visually illustrate the relations among the material. The visualization guides learners and allows them to jump back and forth among the relevant content.

Providing easy access to different conceptual pieces within a lecture video. Previous research has shown that presenting videos along with sub-goals helps people learn better, since the sub-goals can abstract away low-level details and reduce the cognitive load of learners [63, 65, 86]. Since the lecture slides are usually the skeleton of a lecture, and since each slide can be interpreted as a conceptual
piece or sub-goal of this lecture, we designed the interface to visualize the alignment between slides and videos. In this way, videos aligned to different conceptual pieces can be accessed efficiently.

**Minimizing distraction while providing guidance.** The additional navigation and guidance introduce new elements to the interface. Thus learners must learn how the new interaction works, which can be distracting or can lead to cognitive overload. Since distractions have a negative effect on learning [65], we also designed our interface to minimize disturbances. Specifically, we borrowed many design decisions from mainstream MOOC platforms to facilitate intuitive interaction with the interface. Below, we introduce the interface and describe in detail how the design achieves the three goals.

In Fig. 3-7, a screenshot is shown of the interface presenting content and linking simultaneously. In the interface, there are four main components: key-term search, material list, content presentation, and linking visualization. To begin interacting with the interface, the user enters the topic he or she intends to learn in the search field. Our server retrieves the learning materials relevant to the entered topic by 1) stemming the search query for query expansion, 2) enumerating n-grams (n from one to five) in the expanded query, 3) scoring each lecture video, slide, textbook section, and discussion thread with the number of matched n-grams, and 4) returning the material with the $N$ (we set $N$ to 60 in the following experiment) highest scores. Instead of simply presenting the content of the entire course, we provided a search tool in this interface, because search is a common and mature technique that helps users narrow down candidate documents and obtain the desired information.

The returned materials are listed based on their types (i.e., video, slides, textbook, and discussion) and their original position in each material sequence (e.g., chapter index). The material sequence accessed by selecting the "video" tab (i.e., the list of the videos' main titles and subtitles on the left hand side of the screen) is the main flow, or the trunk, of returned content. In addition to videos, we also attach returned slides, textbook sections, and discussion threads (i.e., the leaves) to relevant videos according to the linking annotations; orange, green, and pink icons for these
supplementary contents are appended to the titles of corresponding videos to illustrate the available types of materials. The returned slides, textbook sections, and discussion threads that are not related to any videos are listed under the corresponding tabs next to the "video" tab.

As far as learnability goes, we laid out the interface to resemble the arrangement in prevalent MOOC platforms such as edX. By borrowing the design decisions made by the professional user experience teams in these platforms, we were able to make our interface intuitive, which helps learners to concentrate on learning the content,
instead of learning how to use the website. This fact not only enhances the learning experience, but also reduces noise when we measure how linking affects learning in a user study. In addition, we chose to preserve the original flow of materials when presenting the returned content, instead of listing materials by their relevance scores. Preserving the original flow allows us to visualize the context and prerequisite dependency among returned materials, which is crucial for achieving meaningful learning.

After selecting a video from the list, the learning content along with linking information is shown in the middle. As illustrated in the figure, a lecture video and the synchronized transcription are presented. Under the video scrubber, several orange, green, and pink blocks are rendered. These colored blocks are synchronized with the video progress bar. Each colored block corresponds to a linked supplementary object (i.e., a slide, textbook section, or discussion thread), and the span of the block represents the video vignette that is linked to the underlying object. As shown in Fig. 3-8, by clicking on each colored block the corresponding object is displayed in a lightbox. As for the returned slides, sections, and threads that are not linked to any video, learners can also access the content by selecting the material title from the list under the corresponding tab; the resulting content is also displayed in the middle of the website.

We surmised that this interface would enhance the learning experience by helping learners access relevant information and identify the underlying sub-topics of the videos. As compared to a conventional video player where only the video scrubber is provided, the synchronized object identifiers serve as recommendations that might prove useful for learners at different points in their learning path. For instance, if a learner were watching a lecture video and were confused at a specific point in the video, with our interface this learner could access easily the detailed explanation in the textbook using the linking from the video vignette; if the learner wanted to learn more about a concept mentioned at some point in the video, the linked forum threads could provide further discussion. Furthermore, since relevant materials are linked and placed together under the video scrubber, it is easier to identify redundant learning
Figure 3-8: By clicking on any of the colored blocks under video scrubber, content of the linked supplementary object represented by this clicked block is rendered in a lightbox. In this figure, we provide an example illustrating how a linked slide is displayed in the proposed interface after its corresponding block is clicked.

content, such as duplicated questions in the forum; therefore this interface streamlines navigation.

Additionally, as the lecture slides are typically the skeleton of a lecture, each slide can be seen as a sub-topic or a sub-goal of the lecture. Thus, by aligning slides to a lecture video, we divide the video into several conceptual pieces, where each piece corresponds to a sub-goal or sub-topic. We visualize this alignment in the proposed interface. Hence learners visually identify the structure of the lecture, and navigate to different sub-topics easily.

The remaining part is how to obtain the synchronized object identifiers below the video scrubber. As shown in Fig. 3-9, this can be done easily with the linking annotation described in Section 3.1. With the time code extracted from the video subtitle file, for each supplementary object we obtain the beginning and ending time codes for the segment of transcription sentences linked to this object. The video player with the synchronized linked objects can then be rendered with the time information.
Figure 3-9: With the annotated linking from video segments (i.e., sentences or vignettes) to slides, textbook, and discussions, as well as the time code of each segment, the synchronized linked objects under the video scrubber can be rendered. In this example, each page of slides is indexed with A, B, and C; each textbook section is indexed with α, β, and so on; each discussion thread is indexed with a, b, c, and so on.

### 3.3 Comparative study

To answer the research question "Would it help learners if we were able to link course materials using human annotators?" we assessed the learning effect of linking presented to learners. Specifically, we conducted a comparative study, in which we presented experimental subjects with interfaces with or without linking, and measured their performance on various learning tasks. We focused our study on three aspects:

- How do learners find learning content when linking are provided?
• How does linking affect learners in integrating and memorizing information within a fixed period of time?

• How does linking affect different cohorts of learners?

3.3.1 Study design

We adopted a between-subjects design for our study, where each learner was randomly assigned to either the linking interface (i.e., the interface described in Section 3.2) or a baseline interface without any of the inter-material relations from the linking interface. We introduce the baseline interface in detail in Section 3.3.2.

We designed two learning task scenarios for learners to perform with their assigned interface: information search and concept retention. Learner performance in these tasks was analyzed to investigate the learning effect of linking with respect to the three aspects described above.

• **Information search** tasks involve finding learning content that corresponds to a given problem. In each of these tasks, the learner is randomly assigned a problem sampled from the course quizzes. This learner is then to use the assigned interface to find a piece of learning content that explains how to solve the problem. A piece of learning content can be a specific moment in a lecture video, a page of slides, a textbook section, or a discussion thread (only in 6.00x). This emulates the scenario where the learner is attempting to find informative content to solve a problem.

• **Concept retention** tasks require learners to remember, understand, and integrate concepts relevant to a given topic. In each task, we randomly give a learner a topic sampled from the courses, and allow ten minutes for the learner to learn about the topic using the assigned interface. After the learning stage we ask this learner to write a short essay that includes as many concepts as he or she can remember. In the writing stage, this learner is not allowed to access the interface and learning content. We set a time limit in order to evaluate how efficiently learners browse through the materials, and how efficiently
they capture and remember high-level information. This emulates the scenario where learners attempt to gain an integral and high-level understanding of a topic within a limited amount of time.

Typically, researchers prefer to apply intervention straight in a course, so that they can measure its effects directly. In contrast, in this thesis we choose to focus our investigation around the above two learning scenarios, and explore the navigational behavior of learners, because learning involves complicated mental processes – from motivation and memorization to understanding and problem solving. It may be too unrealistic to expect to ascertain all of these processes in a single set of experiments. Indeed, exploring the effect of linking in a course could introduce many variables and sources of noise, and thus obfuscate any advantage brought by intervention. Therefore, as suggested in previous work [57], we concentrated our investigation on how linking can facilitate material navigation. If we are able to show that linking positively affects this subset of learning processes, there is abundant literature discussing the correlation between navigation and learning [63, 135, 65, 92, 86, 137], thus making self-evident the benefit of linking in learning.

For these two scenarios, we sampled ten problems and topics respectively in each of the two MOOCs (i.e., Stat2.1x and 6.00x) investigated. In this sampling we emphasized the first half of each of the two courses, because lectures from the latter half are typically more advanced, complicated, and require prerequisite knowledge learned earlier in the course. With this emphasis on foundational lectures, we were attempting to reduce noise introduced by the diverse prior knowledge learners may have. In Fig. 3-10, we show two sampled problems from each of the MOOCs along with examples of learning content pieces we accept as answers. In Fig. 3-11, two examples of sampled topics along with one learner’s submission respectively are given. Concepts in these submitted essays are highlighted in bold font. In Appendix A we list the complete sets of sampled problems and topics for each MOOC.
In the following, we show the distribution of midterm scores in a statistic class. Please find the 'inter-quartile range'.

<table>
<thead>
<tr>
<th>midterm scores</th>
<th>percent of students</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-60</td>
<td>10</td>
</tr>
<tr>
<td>60-75</td>
<td>22</td>
</tr>
<tr>
<td>75-85</td>
<td>30</td>
</tr>
<tr>
<td>85-95</td>
<td>13</td>
</tr>
<tr>
<td>95-100</td>
<td>25</td>
</tr>
</tbody>
</table>

The Range, IQR and SD

The three most common measures of spread or variability are the **range**, the **interquartile range (IQR)**, and the **standard deviation (SD)**.

- **The range** of a list is the largest value minus the smallest value. It is the width of the smallest interval that contains all the data, so it measures spread. It is **not resistant** because changing just one datum can make it arbitrarily large.

- **The IQR** is the **upper quartile** (75th percentile), minus the **lower quartile** (25th percentile). It is the width of the interval that contains the middle 50% of the data—and thus it is a measure of spread. It is insensitive to the most extreme values of the data (assuming that there are more than four data). The IQR is **resistant**: changing just one datum has a limited effect on it. Note that neither the range nor the IQR is a range of numbers, despite their names—each is a single number.

- The **RMS** (root mean square) of a list measures the average size of its entries. It is defined as follows:

  \[
  \text{RMS} = \sqrt{\frac{\text{sum of the squares of the entries}}{\text{number of entries}}}.
  \]

What error (if any) is raised when the following code snippets are attempted?

```python
mylist = [10, 20, 30]
mylist.index(11)
```

A: ValueError  
B: TypeError  
C: SyntaxError  
D: NameError  
E: No error is raised

Types of Exceptions

- We've seen the common errors:
  - SyntaxError: Python can't parse program
  - NameError: local or global name not found
  - AttributeError: attribute reference fails
  - TypeError: operand doesn't have correct type
  - ValueError: operand type okay, but value is illegal
  - IOError: IO system reports malfunction (eg, file not found)

- See Section 6 of The Python Standard Library at docs.python.org

3.3.2 Baseline

In the comparative study, we implemented a baseline interface, and investigated whether assigning learners with either the baseline or the linked interface (linking) affected their performance in accomplishing tasks. Thus, we designed the null interface (null).

In Fig. 3-12, a screenshot of the null interface is shown. The only difference in this interface is the lack of the visualization element. It retains linking's visual layouts, as well as the components for key-term search, material list, and content presentation. As illustrated in this figure, in the null interface users also begin by submitting search queries, after which the retrieved materials are also listed according to their types (i.e.,...
The topic you have to learn is: Correlation

Correlation is a measure of linear association: how nearly a scatterplot follows a straight line. Two variables are positively correlated if the scatterplot slopes upwards ($r > 0$); they are negatively correlated if the scatterplot slopes downward ($r < 0$). Correlation is a measure of association, not causation.

The topic you have to learn is: Computational complexity

Computational complexity is a theory that classifies computational problems based on their difficulty. Programmers try to increase a program's conceptual complexity to reduce the computational complexity. Asymptotic notation gives a way to talk about the relationship between running time of an algorithm and its inputs. It becomes less efficient the longer the input. There are many important classes of complexities: constant, logarithmic, linear, log-linear, polynomial, and exponential. Constant is independent of inputs. Log-linear is product of 2 items which are both dependent on the size of the inputs.

Figure 3-11: The first row shows two sampled topics used in the concept retention scenario. For each topic, shown as an example is an essay submitted by a learner in our user study. We also set concepts in essays in bold font. In this figure, the left hand side is a topic-essay pair for Stat2.1x and the right is for 6.00x.

the panels of material types listed in the top left corner) and their original positions in each material sequence (e.g., lecture or chapter indices, as shown on the left hand side of the figure); the learning content selected from the sequence is rendered in the middle. However, every material type is presented independently and no relational information is provided, e.g., the linked supplementary objects (or the leaves) are no longer rendered under the lecture videos (i.e., the trunk). By comparing the linking and null interfaces, we investigate how learner behavior is affected when they are offered information on relations among learning content.

3.3.3 Experiment subjects

For experimental subjects, we chose to recruit online workers from AMT [90]. Generally, we would seek a subject pool of learners who are actually taking the course. However, in our case, online workers are good substitutes for enrolled learners, because we measure their behavior in accomplishing specific learning tasks where clear
Recursion pretty simple.

You're just playing games here. But I'm not. It's a really important point here. I have just reduced this problem to a simple operation and a simpler version of the same problem. And in fact, what is this piece here, that is exactly the same as a times b minus 1. Ah ha.

That's a simpler version of the same problem. I've now thought recursively. Sounds like a highfalutin term. It's really not. It says, I have taken a problem, and

Figure 3-12: The null interface, which serves as one of our baselines. This interface retains linking's components for key-term search, material list, and content presentation. The layout and visual design are also identical. The only difference is that we strip away the linking visualization, and there is no synchronized supplementary learning object under each lecture video (i.e., the trunk). In these tasks, the goals of online workers are very similar to those of learners, such as finding the desired information as quickly as possible or learning more high-level concepts within a limited amount of time. Therefore, they interact with our interface much as enrolled learners do. Besides, although these workers are monetarily driven, as shown in the following quote from a worker’s feedback, financial gain is not their sole motivation.

"I really like this HIT. I hope I am doing them well for you as intended. I want to thank you as well, because I’m actually learning quite a bit about computer programming and I really like the lectures and how they are organized, every time the 10 minutes are up, I’m kind of disappointed because I feel like I was just getting started learning about a subject I’m interested in."

Our HITs also motivate workers intellectually, and attract many who want to learn about the two courses. Moreover, performing a live experiment in an actual MOOC
is expensive and time consuming. In contrast, the abundant online labor pool and the diverse demographics of workers guarantee us access to users of various backgrounds on a large scale, and simultaneously demand a reasonable cost in time and money. This fact allows us to investigate the third aspect – "How does linking affect different cohorts of learners?" – to understand the usefulness of the proposed framework for a heterogeneous learner body. Given these advantages, we thus chose these micropayment online workers as our experimental subjects.

### 3.3.4 Experiment scale

In Table 3.1, the scale of our experiment is summarized. In Stat2.1x, for each scenario we deployed 2,000 HITs on AMT: the null and linking interfaces, 10 problems or topics, and 100 HITs accomplished by 100 unique online workers for each pair of problem/topic and interface. The reward for each HIT was $0.35 and $1.00 for the information search and concept retention scenario respectively. As listed in the table, a total of 497 and 751 unique AMT workers participated in each of the two scenarios. These numbers are different from 2,000 because we allowed each worker to solve more than one problem or topic (under the between-subject design, in contrast, each subject may only work with one assigned interface). The experiment took four months to complete.

The scale of the 6.00x experiment is also shown in the table. Here also, 2,000 HITs were deployed for each scenario. In the two scenarios, 393 and 631 workers participated respectively. Observing the slow completion rate of the experiment in Stat2.1x, we increased the HIT rewards to $1.50 and $2.00 for the two scenarios. In 6.00x, since the relevant topics were more complicated, there were fewer potential and qualified workers for our tasks. This is another reason why we decided to provide larger monetary incentive. This experiment took two and a half months to complete.
Table 3.1: Sizes of comparative studies on Stat2.1x and 6.00x

<table>
<thead>
<tr>
<th></th>
<th>Number of tasks</th>
<th>Number of unique workers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Stat2.1x</td>
<td>6.00x</td>
</tr>
<tr>
<td>Information search</td>
<td>2,000</td>
<td>2,000</td>
</tr>
<tr>
<td></td>
<td>497</td>
<td>393</td>
</tr>
<tr>
<td>Concept retention</td>
<td>2,000</td>
<td>2,000</td>
</tr>
<tr>
<td></td>
<td>751</td>
<td>631</td>
</tr>
</tbody>
</table>

3.4 Results

Given the experimental setup and deployment described above, we then measured subject performance in accomplishing the learning tasks, to explore the effect of linking on learning from three aspects: how does linking affect search, affect integrating or memorizing information, and affect different cohorts. To study how linking affects various cohorts of learners, in our tasks we also required subjects to fill in a background survey. Based on information provided by subjects in the survey, we identified three demographic factors that could influence their performance: their highest accomplished degree, their previous experience in online courses, and their previous exposure to relevant topics/courses.

To study the effect of these factors, we divided subjects with three different criteria (i.e., whether or not they had had exposure to statistics or the Python programming language, whether they had taken MOOCs previously, and did they have at least a bachelor’s degree). In Table 3.2, we list for each Stat2.1x learning scenario (i.e., information search and concept retention) and interface (i.e., null and linking) the numbers of completed tasks classified by criterion. We observe that about seven out of ten and six of ten of the participants in the two scenarios reported prior knowledge in statistics (the second and third row of the table); only about a quarter of subjects in these scenarios had attended MOOCs previously (fourth and fifth rows); slightly more than half of the participants had a bachelor’s or higher degree (sixth and seventh rows). We also break down the completed tasks in 6.00x and summarize the results in Table 3.3. The data show that slightly more tasks were contributed by subjects with previous exposure to MOOCs and with at least a bachelor’s degree, but fewer tasks were completed by subjects with experience in the course topics (i.e., the Python programming language). On top of these divisions, we measured learning performance...
Table 3.2: Number of tasks completed by each cohort for each learning scenario (i.e., information search and concept retention) and interface (i.e., null and linking) in the Stat2.1x study.

<table>
<thead>
<tr>
<th></th>
<th>Information search</th>
<th>Concept retention</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>null</td>
<td>linking</td>
</tr>
<tr>
<td>Overall</td>
<td>1,000</td>
<td>1,000</td>
</tr>
<tr>
<td>Statistics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>714</td>
<td>704</td>
</tr>
<tr>
<td>No</td>
<td>286</td>
<td>296</td>
</tr>
<tr>
<td>MOOCs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>295</td>
<td>249</td>
</tr>
<tr>
<td>No</td>
<td>705</td>
<td>751</td>
</tr>
<tr>
<td>≥Bachelor</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>573</td>
<td>522</td>
</tr>
<tr>
<td>No</td>
<td>427</td>
<td>478</td>
</tr>
</tbody>
</table>

Table 3.3: Number of tasks completed by each cohort for each learning scenario (i.e., information search and concept retention) and interface (i.e., null and linking) in the 6.00x study.

<table>
<thead>
<tr>
<th></th>
<th>Information search</th>
<th>Concept retention</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>null</td>
<td>linking</td>
</tr>
<tr>
<td>Overall</td>
<td>1,000</td>
<td>1,000</td>
</tr>
<tr>
<td>Python</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>455</td>
<td>536</td>
</tr>
<tr>
<td>No</td>
<td>545</td>
<td>464</td>
</tr>
<tr>
<td>MOOCs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>384</td>
<td>397</td>
</tr>
<tr>
<td>No</td>
<td>616</td>
<td>603</td>
</tr>
<tr>
<td>≥Bachelor</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>607</td>
<td>623</td>
</tr>
<tr>
<td>No</td>
<td>393</td>
<td>377</td>
</tr>
</tbody>
</table>

among each cohort to investigate the effect of linking on search behavior as well as integrating and memorizing information.

### 3.4.1 How linking affects search

In this section, we investigate learner performance in the information search scenario. Here, we computed two metrics: average search time and average accuracy. The first metric evaluates how quickly each subject was able to identify a piece of learning content (i.e., a specific moment in a lecture video, a specific slide, a textbook section, or a discussion thread) in answer to the assigned problem and submitted HIT; the second metric measures whether the identified content indeed answered the problem. To measure the accuracy, for each problem the learning content pieces that were valid
Table 3.4: Learner performance in information search scenario in Stat2.1x study. Performance is evaluated by the average search time and average accuracy metrics, and measured within various cohorts using different interfaces.

<table>
<thead>
<tr>
<th></th>
<th>Average search time (seconds)</th>
<th>Average accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>null</td>
<td>linking</td>
</tr>
<tr>
<td>Overall</td>
<td>206</td>
<td>152</td>
</tr>
<tr>
<td>Statistics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>166</td>
<td>147</td>
</tr>
<tr>
<td>No</td>
<td>295</td>
<td>160</td>
</tr>
<tr>
<td>MOOCs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>166</td>
<td>139</td>
</tr>
<tr>
<td>No</td>
<td>225</td>
<td>154</td>
</tr>
<tr>
<td>≥Bachelor</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>198</td>
<td>163</td>
</tr>
<tr>
<td>No</td>
<td>208</td>
<td>136</td>
</tr>
</tbody>
</table>

Answers were labeled. In Stat2.1x, three annotators who were graduate students or postdoctoral researchers with expertise in statistics did the labeling; for 6.00x, we recruited three teaching assistants from the same class offered at MIT to obtain the annotation. Note that when a worker identified a specific moment in a lecture video as the answer, we accepted the submission as correct only if it deviated from any of our labeled answers by less than one minute. With these metrics, we attempted to understand how linking affects learner behavior when they are trying to find learning content.

Table 3.4 summarizes learner performance in the information search scenario in the Stat2.1x study. Performance was evaluated using the average search time (columns 1 and 2) and average accuracy (columns 3 and 4), and measured within cohorts with various backgrounds (row 1 for overall subjects; rows 2 and 3 for whether subjects had prior knowledge in statistics; rows 4 and 5 for whether subjects had attended MOOCs before; rows 6 and 7 for whether they had at least a bachelor’s degree) and using different interfaces (columns 1 and 3: *null*; columns 2 and 4: *linking*). As mentioned above, experiments conducted using online workers are often attended by spammers. To control the quality of worker submissions, in each learner cohort we discarded submissions with search times in the top and bottom 5%. This filtered such as workers trying to cheat the system by randomly selecting a piece of learning content, or workers leaving their computers during tasks.
To examine how providing linking information affected learners in search, we focused on the performance difference between subjects using each of the interfaces. These differences are plotted in Fig. 3-13. For consistency, the length of each bar represents the improvement of a given metric for the linking interface as compared to the null interface. Thus, the upper panel corresponds to the average time using the null interface subtracted by the time using the linking interface. In contrast, the lower panel is computed by subtracting the accuracy when using the null interface from the accuracy when using the linking interface. In the figure, learner cohorts are aligned in the same order as in the table. In addition to the difference values, the 95% confidence intervals are also presented. Furthermore, differences that are statistically significant (we adopted a one-tailed, two-sample t-test for the average search time and a one-tailed, binomial proportion test for the average accuracy; the significance level was set to 0.05) are marked with red asterisk.

Focusing first on row 1 of Table 3.4, as well as the first bar in the upper and lower panels of Fig. 3-13, we see that the overall search time was reduced by 36% (or 54 seconds) when using the linking interface (cf. 206 vs. 152), and that this reduction is statistically significant. In contrast, there was no significant difference in the accuracy between the two interfaces. This shows that subjects were able to find desired information much faster without sacrificing accuracy, and it supports our conjecture that the linking benefits educational content navigation.

Table 3.4 and Fig. 3-13 also include individual results for the three demographic groups. In all six cases the linking interface yielded shorter search times (with reductions from 19 seconds to 135 seconds). Also, this reduction in search times was statistically significant in four out of the six cases: subjects without prior knowledge in statistics, without prior exposure to MOOCs, and with/without a bachelor’s degree or higher. To interpret these results, we classified as naive learners those subjects who were less familiar with the course materials, less experienced with MOOCs, and less educated. This is because subjects with less familiarity must spend more time and effort to catch up on prerequisite knowledge before they can understand a new topic. MOOC learners tend to be self-learners and desire to constantly enrich themselves.
Figure 3-13: Improvement in search time (upper panel) and accuracy (lower panel) when \textit{linking} interface is used. Learning performance improvement was measured in the Stat2.1x study. Also shown are the 95\% confidence intervals (shown as error bars) and significance test results (statistically significant differences marked with red asterisk).

with learning by utilizing any available resource. In contrast, subjects with no experience in MOOCs are more likely to be passive learners or less comfortable learning from online materials. For education, its purpose is not only to teach students specific knowledge, but also to teach how to learn. Thus, there is a higher chance that less educated subjects have less learning experience.

We observe that the \textit{linking} interface yielded greater time reductions for novice subjects. This is perhaps not surprising. As pointed out by Kirschner et al. [65], due to the lack of learning experience and comprehensive understanding of the underlying course topics, novice learners are typically unable to properly explore learning content
on their own. Without providing guidance to these learners, their cognitive system can be overloaded by new topics that must be learned and prerequisite knowledge that must be accumulated; thus, they are more likely to struggle with frustration. As compared to the baseline, our linking interface visualizes linking among pieces of learning content, supports relational navigation among materials, and provides easy access to each sub-goal or sub-concept within a lecture video. These features serve to guide users, helping learners navigate through the learning content. Therefore, learners find information more efficiently, and greater improvement is observed in novices since they are precisely those learners who are more likely to struggle, need more support, and can benefit more from guidance.

As for search accuracy, although the performance difference between the two interfaces varied from -1.4% to 2.2% in various cohorts, none of these discrepancies was statistically significant. Our results indicate that in this experiment linking has little impact on task accuracy. This could be due to the fact that the difference between the two interfaces was in the visualization of inter-material relations; the two interfaces were built on the same set of learning content and search mechanism. Since Stat2.1x is a rather small MOOC that contains only 7-hour lectures spanning 5 weeks, and we used only a limited set of materials (i.e., videos, slides, and textbook) to build this minimum viable product, it was not difficult for patient learners to find the correct pieces of learning content in a reasonable amount of time.

From these results in the Stat2.1x study we conclude that by linking the educational content and visualizing the inter-material relation with the linking interface, learners found desired information more efficiently without sacrificing search correctness. Moreover, among the studied cohorts of learners, novices benefitted more from the provided guidance. These observations show one of the possible ways that linking facilitates learning.

With this encouraging result, we further expanded the horizons of the study by exploring 6.00x, another MOOC. Since learning is dependent on the course subject, with the 6.00x study we attempted to investigate whether the benefit of linking shown above is topic-dependent, or if the improvement is more general. If we were able to
Table 3.5: Learner performance in the information search scenario in the 6.00x study. Similar to the first study, performance is evaluated by the average search time and average accuracy, and measured in various cohorts using different interfaces.

<table>
<thead>
<tr>
<th></th>
<th>Average search time (seconds)</th>
<th>Average accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>null</td>
<td>linking</td>
</tr>
<tr>
<td>Overall</td>
<td>443</td>
<td>349</td>
</tr>
<tr>
<td>Python</td>
<td>Yes</td>
<td>419</td>
</tr>
<tr>
<td></td>
<td>No</td>
<td>463</td>
</tr>
<tr>
<td>MOOCs</td>
<td>Yes</td>
<td>427</td>
</tr>
<tr>
<td></td>
<td>No</td>
<td>454</td>
</tr>
<tr>
<td>≥Bachelor</td>
<td>Yes</td>
<td>472</td>
</tr>
<tr>
<td></td>
<td>No</td>
<td>399</td>
</tr>
</tbody>
</table>

provide evidence showing that linking yielded similar improvements in some learning factors with a different course, it would be stronger proof that the benefit of linking is generalizable to various topics. Additionally, in the 6.00x study we also attempted to evaluate our linking framework in a more realistic scenario. Thus, materials from online forums were also provided in the interfaces; data annotation (the linking among materials and the correct pieces of learning content for each problem in the user study task) was done by teaching assistants instead of the researchers themselves.

Table 3.5 summarizes learner performances in the information search scenario in the 6.00x study. As with the Stat2.1x study, performance was evaluated by the average search time (columns 1 and 2) and average accuracy (columns 3 and 4). These metrics were measured within cohorts of various backgrounds (rows 1 to 7) and using different interfaces (columns 1 and 3: null; columns 2 and 4: linking). We employed similar dividing criteria for the background (i.e., prior knowledge, MOOC experience, and highest degree). Besides, the same quality control mechanism (i.e., discard submissions with search times in the top and bottom 5%) was utilized to minimize noise from spammers.

To examine the benefit brought by linking, here we also focus on the performance difference when the various interfaces were deployed, and plot the differences in Fig. 3-14. Similar to the Stat2.1x study, values of the bars in this figure represent the time reduction (upper panel) and accuracy increase (lower panel) achieved by deploying
Figure 3-14: Improvements in search time and accuracy when linking interface was deployed. Learning performance was measured in the 6.00x study. The 95% confidence intervals and significance test results are also plotted.

the linking interface. Improvement in different cohorts is also displayed in the same order as in the table. In addition we plot the 95% confidence intervals (the error bars) as well as whether the differences were statistically significant (marked with red asterisk if significant).

From Table 3.5 and Fig. 3-14, the first thing we observe is that when the linking interface was deployed, each cohort of experimental subjects took significantly less time to accomplish the tasks. As for the accuracy of the completed tasks, a statistically significant improvement from the linking interface was found in the entire group of subjects, subjects without prior experience in Python language, subjects without prior exposure to MOOCs, and subjects with at least a bachelor’s degree.
No significant difference was measured in the other cohorts.

The observations show that our previous conclusion holds also for a different course: precisely, that when linking is presented, learners find desired information more efficiently without sacrificing search correctness. This result strengthens our claim that linking benefits learning. In addition, we see improvement in search accuracy here in several cohorts. This presumably results from the increased amount of learning materials available. In 6.00x the course was three times longer than Stat2.1x, and the discussions were also available for learners. We hypothesize that when learners have more material to navigate through, being able to visualize the relations between materials has a larger effect on their ability to find information.

To validate this conjecture, we conducted a regression analysis between the search time used in tasks and the accuracy improvement yielded by the linking. It would support our hypothesis if we could find evidence showing that when learners spent longer on their tasks, the linking interface yielded larger accuracy improvements (note that in this study subjects spent twice as much time as that spent in the Stat2.1x study). We designed the regression analysis by first sorting each of the 1,000 tasks in the null group and in the linking group separately according to the search time. Each set of tasks was then divided into 10 equal-sized batches from tasks using the least amount to the most amount of time. We averaged the search time over the i-th batch of the two sets (i.e., tasks using null or linking) as the value of the independent variable of sample i in the regression; the value of the dependent variable we calculated by subtracting the accuracy of the i-th batch of tasks using the null interface from the accuracy of the i-th batch using the linking interface.

The result of our regression analysis is plotted in Fig. 3-15. Clearly our regression model has a positive slope, which shows that search time and improved accuracy are positively correlated. This observation supports our previous conjecture. However, we must note that the p-value of a hypothesis test that the slope is positive is 0.17, which is higher than the usually-used significance level of 0.05.

The second observation that can be made from Table 3.5 and Fig. 3-14 is that both naive and advanced learners benefitted from the linking interface in terms of
Figure 3-15: This first-order regression model (dashed line) relates the average search time (in seconds) of task batches (horizontal axis) to the accuracy improvement yielded by deploying the linking interface (vertical axis).

reduced task completion time. We believe this finding is related to the subject matter in the user study. 6.00x covered a wide range of advanced topics such as algorithms, complexity, computational problem solving, and Python language programming. In contrast to Stat2.1x, the statistics course which contained material typically taught systematically in a high school class, in 6.00x most people are familiar with only part of the topics. For instance, a computer scientist might know algorithms and the theory of computation but might not use Python; a data analyst might be familiar with using Python to analyze data, but might not be an expert in algorithms. Thus, some subjects classified as advanced learners based on our categorization could have been beginners in the topics used in some of the learning task. This may explain why a more uniform improvement over cohorts was observed.

In conclusion, the results in the 6.00x study provide more evidence supporting the benefit of linking in learning. With the linking interface, learners also found desired benefits.

2This can be illustrated with some feedback received in this study: one participant said, "I was already familiar with most of the concepts except for dynamic programming and program complexity. I was a computer science major 30 years ago. Back then they were teaching IBM 360/370 assembly language and FORTRAN 77. I code now in C, Python, PHP, SQL, shell scripts and elisp for my own little projects now and again, although those are too small to warrant much attention for dynamic or complexity considerations." Another participant mentioned: "I already know much about Python, but I find out new things doing these! I don't think I ever really understood the use of recursion until I completed this task."
information more quickly. The improvement in learning performance was observed in both search time and accuracy, and in more cohorts of subjects. These results also highlight the potential of applying linking to various course subjects for better learning. In Section 3.5, we analyze learner click-through patterns in order to provide more understanding and examples about why linking yields better performance; first though, in the next section, we attempt to analyze the benefit of linking from another perspective: information memorization.

3.4.2 How linking affects information memorization

In comparison to finding the desired information, concept retention is a more complicated scenario involving finding, integrating, and memorizing knowledge. In this section we investigate whether linking enhances learners performance in this complex condition. To evaluate the performance we computed one metric: the number of unique key-terms. This metric measures the information richness in paragraphs submitted by subjects, and thus reflects how many concepts relevant to the assigned topic learners can retain after a fixed-length learning stage. We adopted a rather simple metric (as compared to other metrics that evaluate and grade essays [134]) and informed subjects how their submissions would be evaluated, in order to give learners a concrete goal and simplify the learning tasks. From the tasks we attempted to identify factors that were not closely related to material navigation (e.g., the fluency or wording in the essay). Furthermore, other complicated metrics are usually subjective and not easily generalized to different domains (e.g., they require many manually graded essays for an automated grading algorithm to learn from), and thus they do not align with our purpose.

To compute our key-term metric, we needed only label a set of relevant terms for each topic used in the concept retention scenario. For the labeling, we first designated the glossary in each textbook used in Stat2.1x and 6.00x as the set of candidate terms. The same annotators recruited in Section 3.4.1 (i.e., the information search scenario) were also asked to label the relevant topics for each term in the candidate set. With the annotation, we computed the metric simply by counting how many terms were
Table 3.6: Learner performance in the concept retention scenario in the Stat2.1x study. Performance was evaluated by the number of unique key-terms in submitted essays and measured within various cohorts using different interfaces.

<table>
<thead>
<tr>
<th></th>
<th>Number of unique key-terms</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>null</td>
</tr>
<tr>
<td>Overall</td>
<td>4.39</td>
</tr>
<tr>
<td>Statistics</td>
<td>3.98</td>
</tr>
<tr>
<td>Yes</td>
<td>4.71</td>
</tr>
<tr>
<td>No</td>
<td>4.27</td>
</tr>
<tr>
<td>MOOCs</td>
<td>3.98</td>
</tr>
<tr>
<td>Yes</td>
<td>4.83</td>
</tr>
<tr>
<td>No</td>
<td>4.27</td>
</tr>
<tr>
<td>≥Bachelor</td>
<td>3.98</td>
</tr>
<tr>
<td>Yes</td>
<td>4.73</td>
</tr>
<tr>
<td>No</td>
<td>4.27</td>
</tr>
</tbody>
</table>

covered in the essay (since multiple words can sometimes refer to the same word stem, e.g., cats, catty, and cat, in practice we first conducted word stemming [36] to reduce the derived or inflected words in the essay to their word stem before counting the key-terms). This metric allowed us to understand how linking affects learners when they were trying to acquire and remember high-level information about a topic.

Table 3.6 summarizes learner performances in the concept retention scenario in the Stat2.1x study. Performance was evaluated by the number of unique key-terms contained in the submitted essays. As in the information search scenario, the evaluation was also computed within cohorts from different backgrounds (rows 1 to 7) and using different interfaces (column 1: null; column 2: linking). We discovered workers trying to cheat on the tasks by copying and pasting paragraphs found online (e.g., Wikipedia) in their essays. Therefore we utilized an open online plagiarism checker [20]. This checker segmented paragraphs to be checked into sentences, submitted these sentences on Google search, and reported plagiarism if highly similar documents were found on the Web. With this checker, we controlled the quality of the experiment by identifying the spammers and rejecting their results.

To focus on the performance difference when various interfaces were deployed, we also visualize the improvement from the linking interface measured in each subject cohort in Fig. 3-16. That is, the length of each bar represents the average number of unique key-terms when the linking interface was deployed, subtracted by the number
Figure 3-16: The improvement in the number of unique key-terms contained in submitted essays when *linking* interface was used. Learning performance was measured in the Stat2.1x study. The 95% confidence intervals and significance test results are also provided.

when the *null* interface was used. Additionally, the 95% confidence intervals (the error bars) as well as whether the differences are statistically significant (marked with red asterisk if significant) are also indicated. Here, we adopted a one-tailed, two-sample t-test for significance and set the significance level to 0.05.

The first row of Table 3.6 and the first bar in Fig. 3-16 show that, overall, subjects were able to mention a greater number (12%) of key-terms when using the *linking* interface (cf. 4.39 vs. 4.91); furthermore, the difference was statistically significant. Looking over the rest of Table 3.6 and Fig. 3-16, we observe a trend similar to that in the information search scenario, where the *linking* interface yielded improvement over each cohort of subjects, and in four out of the six cases (i.e., subjects with no prior knowledge in statistics, without prior exposure to MOOCs, and with/without a bachelor’s degree or higher) the differences passed the significance test. Furthermore, it seems that novices also benefitted more from linking than advanced learners (e.g., all three naive cohorts show statistically significant improvement).

These results reveal another aspect of the benefit that linking may provide. In the course materials, there were usually many learning pieces relevant to a topic; some complementary pieces, and some redundant. With the visualized inter-material
relation, complementary content could be identified easily and thus be better utilized to reinforce learning. For example, while watching the lecture video, subjects could refer to the aligned slides to understand the lecture at the concept level, as well as to the linked textbook sections or posts for detailed discussions. Identical learning content could also be skipped easily. Furthermore, the visualization helped learners better plan their learning path within the limited-length learning session, and avoid exploring irrelevant or secondary content to the assigned topics. These features made possible by visualizing linking can also be interpreted as the guidance which leads learners navigating through learning content when accomplishing assigned tasks. Therefore, subjects, especially novices, could access knowledge more efficiently in the learning session, and retain more key-terms when they wrote down what they could remember. In Section 3.5 we provide more evidence to support this claim.

We also investigated whether this aspect of benefit can be generalized to various course subjects in a more realistic condition. Hence, similar to the information search scenario, we further studied 6.00x, a different MOOC, and explored it using an expanded material set (i.e., forum discussions were additionally used) and data annotation pipeline (i.e., teaching assistants were recruited as annotators).

Table 3.7 summarizes learner performance in the concept retention scenario in the 6.00x study. Similarly, performance was evaluated by the average number of unique key-terms in the submitted essays, and measured within various cohorts (rows 1 to 7) using different interfaces (column 1: null; column 2: linking). In this study, the same quality control mechanism using plagiarism checking was employed to filter out noise from spammers. Furthermore, to focus on the benefit brought by linking, in Fig. 3-17 we plot the performance difference when the various interfaces were deployed (i.e., number of key-terms when the linking interface was assigned subtracted by that when the null interface was used). In addition to these differences, the 95% confidence intervals (the error bars) as well as the results of the significance tests (marked with red asterisk if significant) are also indicated in the figure.

From Table 3.7 and Fig. 3-17, we first observe that when the linking interface was deployed, subjects in each cohort mentioned a greater number of key-terms. This ob-
Table 3.7: Learner performance in the concept retention scenario in the 6.00x study. As with the Stat2.1x study, performance was evaluated by the number of unique key-terms in the submitted essays and measured within various cohorts using different interfaces.

<table>
<thead>
<tr>
<th></th>
<th>Number of unique key-terms</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>null</td>
</tr>
<tr>
<td>Overall</td>
<td>8.07</td>
</tr>
<tr>
<td>Python: Yes</td>
<td>8.64</td>
</tr>
<tr>
<td>Python: No</td>
<td>7.64</td>
</tr>
<tr>
<td>MOOCs: Yes</td>
<td>8.37</td>
</tr>
<tr>
<td>MOOCs: No</td>
<td>7.93</td>
</tr>
<tr>
<td>≥Bachelor: Yes</td>
<td>8.60</td>
</tr>
<tr>
<td>≥Bachelor: No</td>
<td>7.21</td>
</tr>
</tbody>
</table>

Figure 3-17: The improvement in the number of unique key-terms when the linking interface was deployed. Learning performance was measured in the 6.00x study. The 95% confidence intervals and significance test results are also provided.

...
larger than those in Fig. 3-16.

These results also support our previous claim that when linking is shown in the interface, learners can access information more efficiently and retain more key-terms in their summary of assigned topics. The improvement not only suggests another benefit of linking in learning, but also suggests the possibility of applying our framework in other course subjects. Note that these observations support our hypothesis only that linking is helpful in learning; they do not, however, explain why. Without knowing this, we cannot utilize this linking pedagogy in suitable conditions. Thus, in the next section, we analyze the learner click logs in order to provide an explanation.

3.5 Click log analysis

From the user study results discussed above, we can summarize that by presenting the linking among learning materials, learners were able to access course materials more efficiently and perform better in learning tasks. However, we were also curious to discover how the linking changes the navigational behavior of learners, and why the change yielded improvement in accomplishing our learning tasks. Therefore, we examined the click log generated when learners attempted the tasks. We utilized the log to extract the following three metrics:

- **Number of search queries** used in each learning task.
- **Number of learning objects** surveyed in each task.
- **Time spent** (measured in seconds) in each surveyed learning object.

In each learning scenario (i.e., search and retention), we computed the three metrics averaging over the tasks using each of the two interfaces (i.e., *null* and *linking*). The results are summarized in Table 3.8.

---

3In the 6.00x study, in addition to the submitted answers (i.e., the selected learning object or the essay summarizing the assigned topic), we also recorded how learners interacted with our interfaces. Whenever a learner initiated an event to submit a search query or click on any learning object, that event along with the triggered time was stored in our server.

4Note that because of the way we recorded the event, here the definition of a learning object is slightly different from the rest of this thesis. A learning object in this section refers to a lecture video, a page of lecture slides, a textbook section, or a discussion thread.
Table 3.8: In the information search and concept retention scenario of the 6.00x study, we computed the three metrics (number of search queries used to accomplish a task, number of learning objects surveyed in each task, and the spent time in each learning object) for the two interfaces. The averages ($\mu$) and standard deviations ($\sigma$) of the three metrics are listed here.

<table>
<thead>
<tr>
<th></th>
<th>Information search</th>
<th>Concept retention</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>null</td>
<td>linking</td>
</tr>
<tr>
<td>#Search queries ($\mu, \sigma$)</td>
<td>(2.9, 2.9)</td>
<td>(2.7, 2.8)</td>
</tr>
<tr>
<td>#Learning objects ($\mu, \sigma$)</td>
<td>(10.9, 9.7)</td>
<td>(7.7, 7.5)</td>
</tr>
<tr>
<td>Spent time per object ($\mu, \sigma$)</td>
<td>(32.3, 73.4)</td>
<td>(35.1, 64.9)</td>
</tr>
</tbody>
</table>

Comparing first the numbers in the two scenarios, we observe that in the information search scenario, learners tended to use more search queries, survey more learning objects, and spend less time on each object. Considering the average time learners spent interacting with the interfaces in the two scenarios (6.6 minutes in information search in average and 10 minutes in concept retention), the difference between the two scenarios in the numbers of queries and learning objects is even larger. This discrepancy may derive from the nature of the two scenarios. In the search scenario, learners needed only identify the objects which contained information needed to solve the assigned problems; however, they had to decide which search queries to use. As for the retention scenario, learners were to digest and remember information in the content, but it is obvious that they would use the assigned topics or relevant terms as the queries. Thus, in the search scenario, learners were inclined to survey more queries and learning objects, but spend less time on each of the queries or objects.

We then juxtapose the metrics for each interface within the two scenarios. We find that as compared to using the null interface, when the linking interface was deployed, experimental subjects tended to use fewer search queries (information search: 2.7 vs. 2.9, concept retention: 1.4 vs. 1.6), survey fewer objects (information search: 7.7 vs. 10.9, concept retention: 7.8 vs. 11.8), but spend more time on each object (information search: 35.1 vs. 32.3, concept retention: 70.6 vs. 46.0). We believe this observation explains our user study results. The observation suggests that when linking was visualized, learners were able to identify that learning content which was more informative for the assigned topics or problems; in contrast, when the
null interface was deployed, learners had to submit more queries and access more learning objects. Thus, with the linking interface, learners were able to spend more time understanding the information relevant to the search or retention tasks, yielding better performance. This observation and the user study results can also be related by reduced cognitive load, which has a positive effect on learning [65]: when linking is presented, it is easier for learners to filter out less useful learning objects, which lessens the cognitive load of learners in understanding the materials.

To support our conjecture, we further investigated how relevant the learning objects surveyed by learners were to their assigned tasks. In order to measure this relevance, we utilized the labeled valid learning pieces which were used to evaluate whether the selected learning content was correct in the information search scenario. With the click log recorded in this search scenario, we measured the percentage of surveyed learning objects which contained at least one valid learning piece for the assigned problem.

The mean and standard deviation of the percentage were 0.33 and 0.26 for tasks using the null interface, and 0.50 and 0.33 for tasks using the linking one. We note that when the linking interface was deployed, learners tended to survey more objects containing valid learning pieces. This fact supports our previous claim that when the relations among learning materials were presented, learners were able to filter out learning objects less likely to contain useful content, and focus on informative objects. Thus, better learning outcomes were achieved.

To further illustrate how linking can help learners identify useful information, we visualize two sampled search paths recorded from two subjects when they were completing the assigned tasks using different interfaces. We sampled the two paths based on the following criteria. First, in the paths, subjects surveyed the same number of learning objects as the interface average (i.e., 10 objects for the path recorded in the task using the null interface and 7 objects for the linking one). Second, learners surveyed the same number of objects that contained at least one valid learning piece as the interface average (i.e., 3 informative objects for both interfaces). Third, the two paths were recorded from tasks assigned with the same question.
Samples were taken from a distribution, and the histogram of those samples is shown here:

Which of the following distributions were the samples taken from?

A: Uniform Distribution
B: Exponential Distribution
C: Normal Distribution

Figure 3-18: Question asked in the tasks where we recorded the two sampled search paths.

In Fig. 3-18, we show the question corresponding to the two sampled search paths. This question is about the normal distribution. Figures 3-19 and 3-20 present two search paths observed in tasks using the null and linking interfaces respectively. In these paths, the screenshots and titles of the surveyed learning objects are listed according to the visited order; the material type of each object is also indicated. Furthermore, the titles of objects containing valid learning pieces are set in red; the titles of other objects are in cyan.

From these two paths, we first note that the learner who was assigned the linking interface exhibited significantly better survey quality. Most objects selected by this learner were relevant to the topic of the normal distribution. In contrast, the search path of the learner using the null interface seems unplanned. This observation supports our previous claim that learners benefitted from linking since they were able to
Figure 3-19: The sampled search path recorded when a subject used the null interface to complete the assigned task. In this path this subject surveyed ten objects, three of which contained valid learning pieces. The three objects are indicated with their titles in red; the titles of the remaining objects are set in cyan. These numbers equal the average of the null interface.

more easily filter out less informative materials and focus on the useful ones. Furthermore, we note that the learner using the linking interface switched between different types of materials more frequently. This fact also supports our assertion that when linking is presented, learners are able to utilize complementary information from various types of materials to reinforce their learning.

In this section, we analyzed the click log recorded in the 6.00x user study to explain why linking yields better learning performance in our experiment. We found that when the links among materials were presented to learners, they submitted fewer
Figure 3-20: The sampled search path recorded when a subject used the linking interface to complete the same task as in Fig. 3-19. In this path this learner surveyed seven objects, three of which contained valid learning pieces. These numbers equal the average of the linking interface.

queries and surveyed fewer objects, and spent more time on each object. Moreover, the surveyed objects were more informative; therefore learners performed better on their tasks. However, we should note that the differences in these metrics were not statistically significant. Hence, further stratification of samples is required to obtain stronger evidence.

3.6 Conclusions

This chapter explores our first research question: does manually generated linking help learning? We started by defining two types of linking: homologous and heterologous linking. We formulated the annotation of these two linking types as alignment and binary classification problems respectively, and demonstrated how the annotation can be accomplished by researchers, online workers, and course staff. Then, we imple-
mented a *linking* interface that simultaneously presents learning materials and the annotated linking among them, after which we conducted a large-scale user research study with the two selected STEM courses (statistics and programming languages) to investigate the question.

Our user research showed that the *linking* interface enabled learners to find desired learning content more efficiently and retain more concepts more readily. By analyzing the click log recorded when learners used the interfaces in the user research, we observed that presenting both content and linking at the same time helped learners to focus on informative learning materials, and thus potentially reduced their cognitive loads. These results support the notion that manual linking does indeed improve learning outcomes.
Chapter 4

Can we link automatically?

In this chapter, we investigate methods to link courseware automatically. We showed in the previous chapter that linking helped learners to navigate the course materials, and helped them to find supportive learning content when they were in need or confused. These visualized relations among content also provided guidance that lessened the cognitive loads of learners. Hence the learning experience and outcome were enhanced.

However, from our experience in developing a linking system, annotating relation information requires deep and comprehensive understanding of the course subject, and the labeling process itself is time-consuming. Moreover, even given the linking for the current class, obtaining the relation annotation for future offerings requires much redundant work – since in a new class offering, more forum posts come in (many of which are duplicates), and several lectures may be re-organized. This makes it necessary to maintain and repeat labeling on the updated materials. Thus, manual implementation of an educational content linking framework is cost-prohibitive, inefficient, and not scalable. To facilitate modification of the proposed framework to be more easily deployed and more general, especially given the voluminous amounts of learning materials, we investigate whether the linking of learning content can be generated by machines.

In the following investigation of automated linking methods, we focus on three issues:
• How to design an automated linking method?

• How closely does the automatically generated linking approach human annotation?

• Does automated linking still benefit learners?

For the first issue, we discuss how to formulate linking generation as a prediction/classification problem, and discuss how to solve the problem with machine learning and human language technologies. For the second issue, we compare machine-generated linking with the ground truth established in advance by a human (i.e., the linking we collected in the previous chapter). The comparison is simple, and sheds light on the capability of the implemented automated methods. However, such an evaluation is not perfectly precise, since it measures the similarity between two generated linkings, as opposed to our ultimate goal, which is learning outcome improvement. Furthermore, typically there are many linking configurations which could benefit learners. Computing the similarity to a gold standard is a biased metric which ignores all the other beneficial possibilities. Therefore, we also explore the third issue by conducting a user study on the automated linking, with a pipeline similar to that designed in the previous chapter. Although conducting a user study is costly and time consuming, it evaluates directly the benefits of machine generated linking on learners.

We explored these three issues using the same two MOOCs: Stat2.1x and 6.00x. In our experiments, we found that although there were some differences between the machine- and human-generated linking, they resulted in only slight negative effects on learning. Moreover, the interface driven by automatic linking (below denoted as the auto linking interface) still helps learners to achieve better performance in their tasks. Furthermore, we analyze the difference patterns, and conclude that anecdotally most disagreements between human annotators and our contributed linking algorithm made little difference to learners. In the rest of this chapter, we describe our exploration in detail.
The study discussed above was conducted by comparing an interface presenting linking information to a baseline that implemented the conventional strategy for delivering learning materials online (i.e., the null interface). Since MOOCs are an emerging research field in education, researchers and educators continue to integrate innovative pedagogies into MOOC designs. We are curious how effective our proposed linking framework in comparison to these state-of-the-art techniques. Therefore, in this chapter we describe another user study conducted to compare our linking/auto linking interfaces to the interface currently deployed on edX (denoted as the edx interface). Our results show that learners still achieved better performance in the explored learning tasks with our manually and automatically generated linking.

4.1 Problem formulation

In the design of our automated algorithm we chose to focus on the natural language content in course materials and formulate the linking generation task as a sequential tagging problem. Natural language is an integral part of education for the transmission of knowledge; thus we expected that an algorithm based on the understanding of natural language in learning content would be more generalizable to different courses. The sequential tagging formulation is as follows: we view as a sequence of documents the nodes on the trunk of a linking tree, and represent them as a sequence of input feature vectors \( \mathbf{x} = (x_1, x_2, ..., x_T) \). Determining which supplementary objects should link to these nodes can be interpreted as predicting a sequence of labels \( \mathbf{y} = (Y_1, Y_2, ..., Y_T) \) given \( \mathbf{x} \). Here \( Y_i \) represents the linking configuration of the trunk node \( i \), e.g., the index of an aligned slide or whether a given discussion thread is linked to node \( i \) or not.

We adopted the sequential tagging formulation, because the ordering and context is informative when modeling learning materials. Many theories in the cognitive science of learning suggest that to achieve meaningful learning\(^1\), humans must match

\(^1\)The state where the newly acquired knowledge is fully understood and ready for future use in different circumstances.
the information they learn to how their minds are structured, and integrate this new information with their prior knowledge and existing cognitive system [46, 5]. Guided by these theories, learning content is typically structured in a sequentially dependent manner to help students acquire knowledge. Thus, we surmised that contextual dependency would be helpful in predicting links, e.g., it is likely that neighboring video segments are linked to similar or identical sets of supplementary learning objects.

There are many other applications of this sequential tagging formulation in the natural language understanding domain, including part-of-speech (POS) tagging [104], semantic tagging [70], and machine translation [8]. These applications work at a word-level granularity (i.e., each token is a word) and attempt to interpret the meaning of each token. Since natural language is usually interpreted in sequence by humans, modeling the context is also beneficial in understanding syntax and semantics. In our formulation, a learning object, which can be a video segment, a slide, a textbook section, or a post thread, is adopted as a token, and we model the contextual and lexical dependency upon this larger unit with a similar formulation.

Due to the abundance of applications, many machine learning models have been proposed to solve related problems, including the linear-chain CRF model, hidden Markov models (HMM), general graphical models, and long short-term memory (LSTM) recurrent neural networks [51]. In this thesis, we also adopt linear-chain CRF for our linking problem; HMM can be interpreted as a linear-chain CRF with generative modeling; as compared to linear-chain CRF or HMM, a general graphical model removes the Markov limitation and thus is more complex; LSTM is a model with even higher complexity and non-linearity, and is widely used to express complicated data dependencies.

We chose the linear-chain CRF for several reasons. First, in comparison to many other natural language applications, our corpus was small. This was due to the difficulty of data annotation: labeling the relations among learning objects requires deeper understanding of the content than annotating the POS tags or semantics of

\[^2\text{The linear-chain CRF is a discriminative model.}\]
\[^3\text{Higher complexity means that the underlying model has more free variables and more ability to represent dependency in data.}\]
each word. Our corpus size cannot support the training of complicated models such as LSTM or general graphical models because of the danger of overfitting. Second, as compared to non-linear models such as LSTM, with a linear model it is easier to interpret the results. For instance, each weight in the model can be directly interpreted as the importance of a corresponding feature. This interpretation can facilitate subsequent system development. Third, a discriminative model was preferred in our tasks, since it allows us to incorporate new features into the model without making unnecessary assumptions about the underlying probabilistic distribution\(^4\). This property allows the automated algorithm to be extended easily with various features, which is favorable because there are usually many information modalities in course materials. Below, we discuss how to predict links with CRF under this formulation.

### 4.2 Sequential tagging with CRF

As discussed in Section 3.1, we categorize the relations among learning materials into homologous and heterologous linking, and adopt different annotation methodologies (i.e., alignment and classification). Here we follow the same categorization and design our automated linking algorithm for the two types of relation respectively.

In the homologous case, we applied the linear-chain CRF model to solve the alignment problem. First, we have two types of materials to be linked: one is the trunk and the other is a set of leaf candidates. In this thesis, we designate the trunk as the course's lecture videos; thus the candidates in this homologous case are the lecture slides corresponding to each video. For each lecture \(i\), the video transcription sentences form the sequence of input feature vectors \(x^{(i)} = (x^{(i)}_t)_{t=1}^{T}\) in the CRF (i.e.,

\(^4\)Generative models such as HMM which are based on a full probabilistic model must model the probabilistic distribution of all variables, including both observed (i.e., the features or observations) and unobserved (i.e., the labels) variables. The distribution can be learned from a corpus from scratch but such learning usually requires too many data samples. We can also make assumptions to initialize or limit the distributions to a specific form for the learning to be feasible (especially when we do not have enough data). However, making assumptions is error-prone. In contrast, a discriminative model models only dependency between the observations and the unobserved variables that should be inferred. In such a model we need not learn the entire distribution or make unnecessary assumptions. Therefore, it is much easier to augment a discriminative model with new features (i.e., by introducing new variables and dependencies into the model).
$x_t^{(i)}$ is the $t$-th transcription sentence. Then using CRF, we predict a sequence of labels $y^{(i)} = (y_t^{(i)})_{t=1}^T$ given $x^{(i)}$. Here the value of unobserved variable $Y_t^{(i)}$ is the index of the slide aligned to $x_t^{(i)}$; that is, $Y_t^{(i)} \in \{1, 2, \ldots, S^{(i)}\}$, where $S^{(i)}$ is the number of slides used in lecture $i$. In this way, we transform the alignment task into a problem of inferring the value (i.e., the label or the index of the aligned slide) of $Y_t$ from observation $x$; we then solve this inference problem with CRF.

With CRF’s linear chain structure, the model not only learns the dependence between observation $x$ (i.e., the content of each learning object) and alignment $y$, but also the contextual dependence (i.e., dependence between $Y_{t-1}$ and $Y_t$) over the sequence of objects. Hence, the pattern of order-preserved mapping can be learned during model training. The learned patterns function as probabilistic rules affecting alignment prediction (e.g., if sentence $t - 1$ aligns to slide $s$, it is more likely that sentence $t$ aligns to slide $s$ or $s + 1$, but it is impossible for $t$ to align to slide $s' \in \{1, 2, \ldots, s - 1\}$). These rules model the order-preserving characteristic of homologous linking.

For heterologous linking, we also apply linear-chain CRF to the binary classification problem. Similar to the homologous case, we also have a sequence of learning objects from the trunk (i.e., the lecture videos), and another learning object sequence as the leaf candidates (i.e., textbook sections or forum posts in the following implementation). The video transcription is still the input. However, in contrast to using a sentence as a token, here we adopt a video vignette (i.e., a sequence of sentences aligned to the same page of slides in the previous alignment task) as an item.\footnote{As explained above, we use larger units here to reduce the number of tokens, since each supplementary object is considered separately in this problem, which greatly increases the computation time of the model in training and inference. Furthermore, a video vignette is a more comparable unit to supplementary objects used here, which are either textbook sections or forum posts.} In addition, instead of modeling one sequence of candidate objects at a time, every object is considered separately, and the CRF is used to predict binary labels – whether the considered object is linked or not. That is, in this task the CRF input is $x^{(ij)} = (x_t^{(ij)})_{t=1}^T$ for lecture $i$ and supplementary object (e.g., textbook section or post) $j$; $x_t^{(ij)}$ is the collection of transcription sentences of lecture $i$ which align to
slide \( t \) in the lecture. The model predicts a label sequence \( y^{(ij)} = (Y_t^{(ij)})_{t=1} \), where \( Y_t^{(ij)} \in \{0, 1\} \) encodes whether video vignette \( t \) is linked to supplementary object \( j \) or not.

With this linear chain architecture, in the binary classification task we still model the dependency of linking configurations among neighboring learning objects on the trunk, but with a looser constraint. The model still learns the linking pattern from a sequence of video vignettes. For instance, for neighboring vignettes, they are more likely to share the same relationship (i.e., linked or not linked) to a supplementary object. However, since each supplementary object is considered independently, no dependence across supplementary objects is learned. This modeling of data dependency agrees with our understanding of heterologous linking. As discussed above, heterologous linking is not order-preserving. Thus, certain learning objects arranged closely in one material may imply little about the arrangement of their linked objects in another material. For example, whether learning object \( A_i \) is linked to object \( B_j \) could have little to do with the event that \( A_{i-1} \) is linked to \( B_{j-1} \) due to the variation of object arrangement across materials\(^6\). Since such dependency is of little benefit in predicting linking, modeling the dependency across supplementary and trunk objects simultaneously simply increases the risk of overfitting (due to the increased model complexity) and introduces noise.

In contrast, the linking configuration of a sequence of trunk learning objects to a supplementary one does correlate. This is because topic continuity in educational material is crucial for learners to better digest the content. It is unlikely that the lecturer switches topics abruptly or frequently. Thus, for instance, whether learning object \( A_i \) is linked to object \( B_j \) is dependent on the event that \( A_{i-1} \) is linked to \( B_{j-1} \). In comparison to our model for homologous tasks, this model design is more reasonable for heterologous linking.

\(^6\)Here \( A \) and \( B \) represent two types of learning materials; \( i \) and \( j \) are the indices of learning objects in the two materials respectively.
4.3 Feature extraction

Information in MOOC materials is multimodal. Text, vision, audio content, or even the click log can be useful for linking. To represent the diverse data in a uniform way that can be learned by the CRF model, we must design the feature function set in Equation 2.4. To facilitate label inference, these features should be informative. Below, we discuss the feature design for our linking task.

Lexical similarity features. Since natural language is a central part of education and the transmission of knowledge, we designed our first feature set, lexical similarity features, based on the text content of course materials. These features were designed based on the assumption that the similarity between two learning objects is correlated with whether the two objects are linked. In the alignment task, lexical features can be written as

\[ f_{yk}(Y_t, Y_{t-1}, x) = \cos\text{sim}(\Phi(x_{t+k}), \Phi(y)) \mathbf{1}_{\{Y_t = y\}}, \quad k \in \{-K, -K+1, \ldots, K\} \text{ and } y \in \{1, 2, \ldots, S\} \]  

(4.1)

where \( \mathbf{1} \) is an indicator function and \( K \) is a hyper-parameter deciding the length of context considered in the model. \( \cos\text{sim}(x_{t+k}, y) \) is the cosine similarity between the vector representation (defined by \( \Phi \)) of video transcription sentence \( t + k \) and the supplementary learning object \( y \). In the alignment task, \( y \) is the page index for lecture slides. Thus learning object \( y \) is the \( y \)-th page in the slides.

For the binary classification task, the lexical features we extracted are

\[ f_k(Y_t, Y_{t-1}, x) = \cos\text{sim}(\Phi(x_{t+k}), \Phi(lo)), \quad k \in \{-K, -K+1, \ldots, K\}. \]  

(4.2)

Since in this task each supplementary learning object is considered separately, the lexical features compute the cosine similarity between the vector representation of this supplementary object \( lo \) and video vignette \( t + k \) (i.e., sentences aligned to slide \( t + k \) in the previous task).

Computation of the cosine similarity requires that we define the vector representation, \( \Phi \), of a document (i.e., a video sentence, a video vignette, or a supplementary
learning object). Our first adopted vector used a bag-of-words (BoW) representation. In this representation, we compute the TF-IDF score of each word in the document and transform the document to a vector where each dimension corresponds to the score of a unique word. Second, we adopted the word2vec representation, which was introduced in detail in Section 2.3.3. Word2vec is a continuous language model trained with a neural network to compute the word probability based on the word’s context in the corpus. After the model is trained, each word is represented as a vector in a continuous space by collecting the neural network weights corresponding to that word. With this word-level embedding, each word in a document is transformed first to its word2vec representation, and the document vector is computed by averaging these word vectors. In contrast to the BoW model, the word2vec embedding supports the learning of long-term semantic and syntactic regularities in language. We believe that our linking algorithm can understand learning objects from different aspects using these two vector representations.

**Transition features.** As discussed above, we sought to learn the contextual dependence of the linking configuration with CRF. Thus, we designed the second feature set – transition features:

$$f_{yy'}(Y_t, Y_{t-1}, x) = 1_{\{y_t=y\}}1_{\{y'_{t-1}=y'\}}, \ y, y' \in Y$$

(4.3)

where $Y$ is the set of labels. The assumption behind these features is that the inference of a linked object for two consecutive video segments (i.e., sentences or vignettes) is dependent. These features are typically used in applications of CRF to encode temporal dependencies; here they allow our CRF model to learn temporal patterns of the linking configuration.

**Visual features.** Lecture videos are usually the center of MOOCs. In addition to the human language content such as video transcription, the visual channel also provides rich information with which to understand the materials and infer linking. For example, studies show that scene changes in educational videos affect the watching behavior of learners and usually coincide with structural breaks in videos [64]. Hence,
we designed a set of visual features to extract this useful information:

\[ f_y(Y_t, Y_{t-1}, x) = \text{frame\_distance}(t)1_{\{y_t=y\}}, \quad y \in Y. \] (4.4)

Here we define frame\_distance(t) as the Euclidean distance between video frames corresponding to the beginning and end of video segment t. The time code information of sentences is encoded in the video subtitles, which are typically provided in MOOCs to enhance material accessibility. If there are no subtitles, we obtain the time code by aligning the audio signal with the lecture transcription or by performing automatic speech recognition.

Since video frames are represented by the color of each pixel, we must also transform this information to vectors to computing the distance. We investigated two vector representations: the HSV (hue, saturation, and value) histogram and the horizontal projection. The HSV histogram is widely used in tasks such as scene detection, and represents a frame by its color distribution \[11^8\]. The HSV histogram of a video frame is obtained by first transforming the RGB color value of each pixel in the frame to the HSV space. Then the three coordinates (i.e., H, S, and V) of the HSV space are discretized into a number of bins. The number of pixels in each bin is counted to compute a three-dimensional histogram, and the histogram is flattened to a vector as the HSV representation of the frame. The HSV histogram is popular since it models the mechanism of human color perception. Furthermore, as opposed to simply representing frames as vectors of color of each pixel, the histogram method is much more robust to noise.

However, the HSV representation fails to capture some distinct characteristics of educational videos. For instance, in MOOCs, it is usually the case that the entire video consists of shots of slides or scenes with very similar colors. The HSV descriptor cannot effectively distinguish among these slides or scenes. Thus, we implemented horizontal projection, the second descriptor. To extract this descriptor, for a frame with \(m\) by \(n\) pixels in the HSV space, we first represent the frame with three \(m\)-by-\(n\) matrices corresponding to the three coordinates (i.e., H, S, and V). For each matrix
we add up the intensity of each row to project the visual content along the horizontal direction and obtain an $m$-by-1 vector. The horizontal projection of a frame is then the concatenation of vectors from the three matrices. Since much informative content in educational videos is presented horizontally (e.g., bullets in slides), this tailored descriptor describes video frames in a more pedagogically meaningful way.

One thing that should be noted is that in Equation 4.4, these features depend only on the frame distance and label of segment $t$. This is problematic since the frame distance between the beginning and end of sentence $t$ has little to do with the label of the sentence, but is highly dependent on whether the labels of sentences $t$ and $t-1$ are different. With the current label set, the dependence between label transition and frame distance cannot be represented by these features.

Thus, we added boundary, another label, to the original label set. In Fig. 4-1, we illustrate how this additional label works by comparing the same linking configuration represented with two label sets. The upper panel of Fig. 4-1 corresponds to the original label set described above, where the value of the label is the page index of the linked slide (alignment task) or whether the considered supplementary object is linked (binary classification task). In the lower panel, an additional boundary label (represented as 'bnd' in the figure) is also employed to denote the case where the two consecutive segments are linked to different slides or have different relations to the considered object. With the additional boundary label, the dependence between label transition and frame distance can be encoded in the visual feature functions, and the extended label set has the same ability to express linking configuration as the original set. Thus, we adopted the extended label set for incorporating visual channel information to our automated linking algorithm.

Another possible solution to encoding the dependence between label transition and frame distance would be to use feature functions that depend on both $Y_t$ and $Y_{t-1}$. However, this solution would increase the number of features and make the model more likely to overfit when a small training corpus is used. Thus we chose the extended label set instead.

In this thesis, we only use the frame distance features to encode visual information
Figure 4-1: The same linking configuration represented with two different label sets. In the upper panel, the original label set described in Section 4.2 is used. A, B, and C denote the aligned slide index; Y and N denote whether the considered object is linked. In the lower panel, the original label set along with a boundary label (denoted as 'bnd') is used.

in course materials. There are other methods to understand the visual channel, e.g., optical character recognition (OCR), and semantic understanding of images [45]. However, OCR provides little additional information that could supplement other resources such as slides or video transcription. Current image semantic understanding extracts only shallow information such as there is a man writing on the black board or the woman is talking. Thus, although these methods are gaining in popularity, we believe they are not suitable or mature enough for our tasks, and chose not to investigate them here.

Metadata features. Descriptions of the learning content also can facilitate linking inference. Thus we extracted metadata features to encode such information. In this thesis, two types of metadata features were used: position and learner tagging. Below we describe the two features respectively.

The following is the position features equation in the alignment task:

$$f_y(Y_t, Y_{t-1}, x) = \exp\left(\frac{t}{T} - \frac{y}{S}\right), \quad y \in \{1, 2, ..., S\}. \quad (4.5)$$
This feature set encodes the difference between the relative position of a video transcription sentence and a slide in their original sequence. The relative position is computed by dividing the index of the sentence or slide by the number of sentences or slides in the lecture. The motivation of this feature set is self-evident: if a sentence is mentioned at the beginning of the lecture, this sentence is more likely to be aligned to the first couple of slides.

We also extracted similar features for the binary classification task:

\[ f(Y_t, Y_{t-1}, x) = \exp\left(\frac{t}{T} - \frac{\text{index}(lo)}{N}\right). \]  

(4.6)

In this case, \( t \) is the video vignette index in the entire sequence of lecture videos, and \( T \) is the number of video vignettes in the entire course. \( \text{index}(lo) \) is the index of the supplementary learning object under consideration, and \( N \) is the number of supplementary objects in the course (e.g., the number of textbook sections or discussion threads). The textbook and forum is sorted and indexed based on the section number and thread creation time respectively.

We also used learner tagging as a set of metadata features. In the MOOC platform where we collected learning content for experiments in this thesis, learners were allowed to post discussions under each lecture video [32]. We recorded this tagging information and created a function \( lo(t) \) which maps the \( t \)-th video vignette to a set of discussion threads which were posted under the video this vignette belongs to. With this function, we extracted our learner tagging features as

\[ f(Y_t, Y_{t-1}, x) = 1_{\{lo \in lo(t)\}}, \]  

(4.7)

where \( lo \) represents the discussion thread we considered each time. Although learners sometimes post irrelevant discussions under the video, such as chats with fellow learners, we still believe this feature set is helpful for our automated linking system because it narrows down the number of vignettes that can link to each thread. Note that this tagging information is only available in discussions on the current MOOC platform; thus the learner tagging features are only deployed in the task of linking.
As described above, we observe that features we utilized here are diverse in their forms and value ranges. For all these feature functions, we only assume the existence of dependence between certain observations and labels; no assumption of distributions for these dependencies is required. Thus, in comparison to a generative method such as HMM, with our CRF models it is easier to add new features and extend the automatic linking algorithm.

4.4 Evaluation: similarity to human labeling

We then evaluated our linking algorithm on material in the two MOOCs introduced in Section 2.4: Stat2.1x and 6.00x. In the Stat2.1x MOOC, we had lecture videos, slides, and a textbook; for the 6.00x MOOC, we used the previous three types of materials along with forum posts. Thus, we investigated automated homologous linking between lecture videos and slides in the two MOOCs. For the heterologous case, we studied video-to-textbook-section linking in Stat2.1x, and video-to-textbook-section along with video-to-discussion-thread linking in 6.00x. Since in the heterologous case the video vignette was used as the linking unit, in the experiment we used a two-pass procedure: we first trained a CRF to align video sentences to slides. Then we utilized the best alignment result in the development set to obtain the video vignette (i.e., define the sentences that align to each slide as one video vignette). With these video vignettes we then trained CRFs to predict linking between vignettes and textbook sections or forum discussions.

As discussed at the beginning of this chapter, in the evaluation we explored two issues: 1) how closely does the automatically generated linking approach human annotation? 2) Does automated linking benefit learners? In this section, we investigate the first issue. Here we compare automated linking results to human labeling as collected and discussed in Section 3.1.4, and compute F1 scores to measure the similarity. We explore the second issue in the next section.

In the evaluation, instead of simply splitting the corpus into training and testing
Table 4.1: The F1 scores (%) of automated linking systems in Stat2.1x using different models (logistic regression and CRF) as well as lexical (BoW stands for bag-of-words and word2vec for the neural network word embedding) and visual (HSV=HSV histogram and HP=horizontal projection) features. Performance of both homologous (i.e., linking between video sentences and slides) and heterologous (i.e., linking between video vignettes and textbook sections) tasks is listed. In the table, the parentheses after word2vec denote that the HP visual features were deployed only in the homologous task.

<table>
<thead>
<tr>
<th>Linking systems (model, feature)</th>
<th>Homologous</th>
<th>Heterologous</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logistic regression, BoW</td>
<td>74.9</td>
<td>29.6</td>
</tr>
<tr>
<td>CRF, BoW</td>
<td>81.3</td>
<td>45.2</td>
</tr>
<tr>
<td>CRF, BoW + HSV</td>
<td>84.9</td>
<td>44.6</td>
</tr>
<tr>
<td>CRF, BoW + HP</td>
<td>85.8</td>
<td>44.8</td>
</tr>
<tr>
<td>CRF, word2vec (+ HP)</td>
<td>86.2</td>
<td>45.1</td>
</tr>
<tr>
<td>CRF, BoW + word2Vec (+ HP)</td>
<td>86.7</td>
<td>47.2</td>
</tr>
</tbody>
</table>

sets, we adopted a 5-fold cross validation technique. Specifically, we partitioned our materials into five equal-sized batches. Every time, we chose one batch for testing. The remaining four batches were used for training and hyper-parameter selection. We iterated the training-testing procedure five times with different batches as test sets, and averaged the test-set F1 scores to evaluate the model performance. We adopted a cross validation technique here, because we wanted to present the entire course with machine-predicted linking. If we split the corpus into training and testing, the machines would only predict linking in a portion (i.e., the test set) of the courseware - it would be meaningless to predict linking in the training set since information from the human annotation was already used to train the model.

We first investigated the performance of machine-generated linking in Stat2.1x. In Table 4.1 we summarize the F1 scores of the automated linking systems using the various models and features in the homologous (i.e., linking between video sentences and slides) and heterologous (i.e., linking between video vignettes and textbook sections) tasks. To obtain comparable evaluation metrics, in both tasks F1 scores were computed at the sentence level. Therefore in the heterologous case, before computing the F1 scores, we first mapped the vignette-level linking results to the sentence level.

In this table we study how visual and lexical features affect linking performance by
using various vectorization techniques to encode learning objects for computing video frame distances and lexical similarities. Here, HSV histograms (denoted as HSV) and horizontal projections (denoted as HP) descriptors were investigated to represent video frames; bags of words (denoted as BoW) and neural network embeddings (denoted as word2vec) were studied to represent text content. Since text content is so central to education for the transmission of knowledge, and because BoW is the most widely used text vectorization in natural language processing, in our study we started with linking systems using BoW for lexical features. The word2vec and visual features were gradually introduced to the systems to investigate their potential benefits.

In addition to visual and lexical features, we also evaluated the benefit of using transition features (i.e., modeling contextual dependencies) in predicting linking. We studied the effect of transition features by comparing the CRF algorithm with a baseline model: logistic regression. In logistic regression, the alignment and linking of each video segment were predicted separately and no contextual dependency was modeled. Thus by comparing logistic regression and CRF systems using identical visual and lexical feature sets, the potential benefit of transition features in linking performance could be studied.

As for the metadata features, we utilized them as default features in every linking system we reported, since these features have been shown to be beneficial in general in a variety of applications [49, 117]. In the experiment here with Stat2.1x, positional features were the metadata features deployed in the linking system.

In this table first we see that the logistic regression model (row 1) performed significantly worse than other systems. When an identical feature set (except transition features) was used, CRF outperformed logistic regression by 6.4% in homologous linking and 15.6% in the heterologous task (c.f., row 1 and 2). This observation shows the benefit of formulating linking as a sequential tagging problem: by treating the video segments as a data sequence and introducing transition features into the system, we model the temporal patterns of the linking configurations. In contrast to simply modeling the content similarity in materials such as in logistic regression, the
contextual information of the CRF allows it to better understand how the topics in courseware are organized and dependent on one another. Therefore, in both tasks transition features yield better linking performance.

On top of the lexical features computed from BoW embedding, we investigated how the additional visual features affected the performance of CRF linking systems (c.f., rows 2 to 4). In homologous linking, both HP and HSV features yielded improvement. In order to explain why visual features were helpful in this task, we analyzed our course materials, finding that lecture videos are usually dominated by colloquial speech, and about only 23% of the sentences in the video transcription contain key-terms (terms that appear in the textbook glossary) which are useful to identify the underlying concepts. This small percentage shows that lexical information is sparse. In addition, the lexical similarity features are sometimes noisy at the transition of two slides or topics. For instance, the lecturer could conclude the previous topic or connect two topics using a story. Thus, information from verbal content is often insufficient to infer linking. In contrast, the forte of our visual features is encoding information pertinent to scene changes, which can provide complementary clues to the lexical features for use in aligning slides and videos. Thus, combining both features yields better performance. Comparing rows 3 and 4, we further find that HP outperformed HSV in linking performance. We believe that HP is more suitable for our tasks since it is tailored to educational videos where many contents are presented horizontally.

As for the heterologous task, we find that neither visual feature yielded any improvement. This is presumably because information encoded in these visual features mostly concerns scene changes, and offers little benefit in inferring the links between textbook sections and the video transcription. Also, whatever useful information the visual features might have, has very likely already been encoded in the alignment we used to obtain video vignettes. Thus, it appears that visual features in this task simply introduced irrelevant or redundant information to the model, and thus no improvement was observed.

We then studied the benefit of the word2vec embedding on top of our current best
systems (i.e., row 4 in the homologous task and row 2 in the heterologous task). In row 5 we replaced the BoW lexical similarity features with the word2vec features. In the table, the parentheses after word2vec denote that the HP visual features were only deployed in the homologous task. As compared to the previous best systems, similar linking performance was achieved (c.f., 85.8% to 86.2% in the homologous task and 45.2% to 45.1% in the heterologous task). In row 6, we integrated the lexical features computed from both BoW and word2vec embeddings, yielding further performance improvements (c.f., 86.2% to 86.7% in the homologous task and 45.1% to 47.2% in the heterologous task).

To explain these improvements, we analyzed the two lexical feature sets and found that values of similarity features computed from the BoW embedding were sparse, i.e., most values were zero. This is presumably because each word in BoW representation is treated as an atomic unit; each word's corresponding element in the vector works independently from others. Thus, if two documents have no overlapping words, the cosine similarity between them is zero. Similarity features computed from this embedding encode information about how many overlapping words two learning objects have, which is highly correlated with whether the two objects are linked. However, sometimes different terms are chosen to express the same meaning in different objects or materials. Under such conditions, the BoW embedding fails, tending instead to produce false negatives, i.e., two relevant objects whose similarity is zero or underestimated.

For the word2vec embedding, however, we found that the computed similarity values were much smoother. In the word2vec representation, words and documents are represented as vectors in a continuous space, where dimensions work jointly to encode different semantic or syntactic regularities. Semantically or syntactically related words could be distributed closely in the continuous space, and it is possible for the meaning of unseen words to be reconstructed given their relevant terms. This makes up in part for discontinuities caused by mismatched wording between two documents. However, the encoding yields more false positives, i.e., two irrelevant objects whose similarity is high. Based on these findings, we believe that the BoW and word2vec
Table 4.2: F1 scores (%) of automated linking systems in 6.00x using different models (logistic regression and CRF) and lexical (BoW and word2vec) and visual (HP) features. Performance of both homologous (i.e., linking between video sentences and slides) and heterologous (i.e., linking between video vignettes and textbook sections/discussion threads) tasks is listed.

<table>
<thead>
<tr>
<th>Linking systems (model, feature)</th>
<th>Homologous</th>
<th>Heterologous</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Videos to slides</td>
<td>Videos to textbook</td>
</tr>
<tr>
<td>Logistic regression</td>
<td>BoW</td>
<td>63.3</td>
</tr>
<tr>
<td>CRF</td>
<td>BoW</td>
<td>71.7</td>
</tr>
<tr>
<td></td>
<td>BoW + HP</td>
<td>73.7</td>
</tr>
<tr>
<td></td>
<td>BoW + word2Vec (+ HP)</td>
<td>74.7</td>
</tr>
</tbody>
</table>

embeddings provide complementary encodings of text content; therefore integrating similarity features computed from the two yielded better performance.

We then applied this method to generate the 6.00x linking. Table 4.2 lists the performance of automated linking systems using different models and features. Here we also investigated the linking between video sentences and slides for the homologous linking; for heterologous task, in addition to the linking between video vignettes and textbook sections, linking between vignettes and discussion threads was also studied. Similar to the Stat2.1x experiment, we adopted the sentence-level F1 score as the evaluation metric.

In this thesis, we utilized 6.00x to investigate the generalizability of the proposed methods. Thus, in this experiment we explored only the system configurations which showed improvement in the Stat2.1x experiment to determine whether these improvements could be generalized to different courses and materials. Specifically, we studied three techniques shown to be helpful above: 1) modeling contextual dependency with transition features, 2) adding visual features to detect scene changes in the alignment task, and 3) integrating complementary lexical similarity features computed from both BoW and word2vec embeddings. Here, we also started with systems using metadata features and lexical features from the BoW embedding, and added other features incrementally for the investigation.

---

7Position features were used in all three linking tasks: video segments to slides, textbook, and discussions. The learner tagging features were also used to linking video segments and discussions.
In Table 4.2 we observe that the three techniques again yielded improvement. Comparing rows 1 and 2, CRF outperformed logistic regression consistently in the three tasks (8.4% in linking videos to slides, 3.3% in linking videos to textbook, and 0.8% in linking videos to discussions). This improvement shows that the benefit of modeling contextual dependency generalizes over these tasks. However, improvements in linking between videos and discussions were relatively small as compared to other tasks. We believe this is because relations between videos and discussions are distinct from videos and others. Lecture videos, slides, and textbook are created by the educators and represent a systematic attempt to transmit knowledge. In contrast, most content in the forums is created by learners to resolve specific sources of confusion. The topic organization of learner-created materials is very different from that of educator-created materials. Hence contextual dependency features, which are designed to improve linking prediction via an understanding of the topic organization in materials, were much less helpful when linking videos to discussions.

Comparing rows 2 and 3, we find that visual features also yielded improvement in this programming course. Note that here we only explored the horizontal projection descriptor in the homologous task, since in the Stat2.1x experiment, HP yielded better performance in the homologous task, and none of the visual features were helpful in the heterologous task. In comparison to the results observed in Table 4.1, we find that the performance improvement yielded by HP was smaller (71.7% to 73.7% here and 81.3% to 85.8% previously). We surmise that this is because of the differences in the video styles used in the two courses. We observe in Stat2.1x that a large portion of the lecture video was simply shots of slides; however in 6.00x there was also a great deal of live coding demos and talking head sessions. The demos and talking head sessions introduce noise into our visual features, complicating the detection of slide changes. From these results, we conclude that the benefit of the visual features can be generalized, but the degree of improvement yielded depends on the style of the underlying lecture video.

We then further integrated the lexical features computed from the BoW and word2vec embeddings. In the table we observe that the combination of the two em-
beddings again enhanced the linking performance consistently (c.f., 73.7% to 74.7%
in linking videos to slides, 69.3% to 71.1% in linking videos to textbook, and 32.1%
to 33.3% in linking videos to discussions). These results imply that the complement-
tarity of BoW and word2vec embeddings could be a general fact found in different
courseware; thus the integration could yield generalized improvement over different
courses.

From Tables 4.1 and 4.2 we observe widely-varying F1 scores (the best result
of each linking task ranged from 33.3% to 86.7%). To investigate why this is, in
Table 4.3 we compare the best F1 scores to the annotator agreement (described in
detail in Section 3.1.4) for each linking task. The first two rows summarize the
best performance of our automated linking systems in F1 scores, and the following
two rows list the labeling consistency among annotators evaluated in terms of kappa
scores. The kappa scores can be interpreted as a measurement of how difficult and
ambiguous the underlying linking task was for humans. In this table we find that the
machine performance is highly correlated with the ambiguity of the linking task. This
finding is reassuring, especially for the video-to-discussion task where only a 33.3% F1
score is achieved, since a significant portion of difference between the machine- and
human-labeled linking could be attributable simply to the underlying task ambiguity,
which detracts much less from the learner experience than linking irrelevant learning
objects.

In this section, we showed that the CRF-based linking method integrates inform-
from different features and yields better performance than the conventional

Table 4.3: Best performance of proposed automated linking system (evaluated with
F1 scores, listed in first two rows) and annotator agreement (evaluated with kappa
scores, listed in third and fourth rows) in each linking task

<table>
<thead>
<tr>
<th></th>
<th>Homologous</th>
<th>Heterologous</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Videos</td>
<td>Videos</td>
</tr>
<tr>
<td></td>
<td>to slides</td>
<td>to textbook</td>
</tr>
<tr>
<td>Automatic linking</td>
<td>Stat2.1x</td>
<td>86.7</td>
</tr>
<tr>
<td>(F1 scores, %)</td>
<td>6.00x</td>
<td>74.7</td>
</tr>
<tr>
<td>Annotator agreement</td>
<td>(Kappa scores, %)</td>
<td>86.5</td>
</tr>
<tr>
<td></td>
<td>Stat2.1x</td>
<td>81.0</td>
</tr>
<tr>
<td></td>
<td>6.00x</td>
<td></td>
</tr>
</tbody>
</table>
logistic regression method. By modeling contextual dependency and combining the complementary BoW and word2vec embeddings, we observed consistent improvement in all the linking tasks studied. For visual features, we found they were only helpful in homologous tasks. In these experiments, we showed the extensibility of CRF to different features. We believe this characteristic fits our problem well. Given the recent popularity of both MOOCs and machine learning research, the styles of learning materials and algorithms for understanding content are ever-changing. Given our model's extensibility, it can grow with new machine learning techniques, pedagogies, and content by adding the corresponding features (e.g., our CRF can use as features even posterior probabilities or classification results predicted by a neural network). In addition, the improvement across courses and materials to some extent demonstrates the generalizability of the proposed method, which is crucial for our framework to be widely applied in different conditions.

In the previous discussion we also examined the correlation between model performance and annotator agreement. We surmise that it is likely that the low F1 scores will not greatly detract from the learner experience. In the next section, we attempt to provide evidence for this conjecture with a user study experiment.

4.5 Evaluation: benefit in learning

We then investigate the second question of the evaluation: do automatically-generated linking still benefit learners? To answer this, we conducted a user study similar to that in Section 3.4. We studied learner performance when course materials were presented with various strategies: different types of materials presently separately (i.e., presented in the null interface) or linked by humans (i.e., linking) or by machines (i.e., auto linking). After replacing manually-labeled linking with those generated by machine, if we were to observe performance improvements similar to that described in Section 3.4, we could conclude that our automated linking system is beneficial to learners. Below, we also study the effect of automatic linking on learners in the search

8In this section, Section 4.4's best results are used in the interface for the user study.
4.5.1 How automatic linking affects search

Here we investigate how automated linking affects learners performance in the information search scenario. Performance was also evaluated with two metrics: average search time and average accuracy. Table 4.4 summarizes the Stat2.1x performance. The results of the null and linking interfaces are identical to those discussed in Section 3.4.1. Here, auto linking corresponds to the condition where learners accomplished the assigned tasks using an interface that presents automatic linking. Other than the deployed interfaces, other experimental procedures in the auto linking study were identical\(^9\). Moreover, to examine how automatic linking benefits learners, we also focused on the differences in performance yielded by the various interfaces. Fig. 4-2 visualizes the improvement from linking (red bars) and auto linking (black bars) as compared to null. The upper panel corresponds to the time reduction yielded by the linking and auto linking interfaces in different learner cohorts; the lower panel shows the accuracy increase from the two interfaces. The 95% confidence interval of the difference is also provided.

Similar to the linking interface, in Table 4.4 and Fig. 4-2 we observe that an interface driven by machine-generated linking also helps learners find information within a shorter period of time in general; as for the search accuracy, no statistically significant difference was found. Among the studied learner cohorts, novice learners (subjects without prior knowledge in statistics, without prior exposure to MOOCs, and without a degree higher than bachelor’s) as well as subjects with a degree higher than bachelor’s showed statistically significant improvements in search time reduction. When we compare the improvements yielded by linking and auto linking, we find that in each cohort subjects using the linking interface consistently needed less time in the

\(^9\)We also collected 1,000 HITs for the same 10 questions on the Amazon Mechanical Turk. Since a between-subjects design was used, only online workers who did not participate in our experiment with the null and linking interfaces were allowed in the study. The auto linking study was conducted three months after the experiment on the null and linking interfaces was complete. In addition, the same quality control mechanism was applied here.
Table 4.4: Learner performance in the information search scenario of the Stat2.1x study. Performance was evaluated by the average search time and average accuracy metrics, and measured within various cohorts using the null, linking, and auto linking interfaces.

<table>
<thead>
<tr>
<th></th>
<th>Average search time (seconds)</th>
<th>Average accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>null</td>
<td>linking</td>
</tr>
<tr>
<td>Overall</td>
<td>206</td>
<td>152</td>
</tr>
<tr>
<td>Statistics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>166</td>
<td>147</td>
</tr>
<tr>
<td>No</td>
<td>295</td>
<td>160</td>
</tr>
<tr>
<td>MOOCs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>166</td>
<td>139</td>
</tr>
<tr>
<td>No</td>
<td>225</td>
<td>154</td>
</tr>
<tr>
<td>≥ Bachelor</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>198</td>
<td>163</td>
</tr>
<tr>
<td>No</td>
<td>208</td>
<td>136</td>
</tr>
</tbody>
</table>

Table 4.5: Learner performance in the information search scenario in the 6.00x study. Performance was evaluated by the average search time and average accuracy metrics, and measured within various cohorts using the null, linking, or auto linking interfaces.

<table>
<thead>
<tr>
<th></th>
<th>Average search time (seconds)</th>
<th>Average accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>null</td>
<td>linking</td>
</tr>
<tr>
<td>Overall</td>
<td>443</td>
<td>349</td>
</tr>
<tr>
<td>Python</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>419</td>
<td>323</td>
</tr>
<tr>
<td>No</td>
<td>463</td>
<td>378</td>
</tr>
<tr>
<td>MOOCs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>427</td>
<td>336</td>
</tr>
<tr>
<td>No</td>
<td>454</td>
<td>357</td>
</tr>
<tr>
<td>≥ Bachelor</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>472</td>
<td>359</td>
</tr>
<tr>
<td>No</td>
<td>399</td>
<td>331</td>
</tr>
</tbody>
</table>

search task.

We also explore how automated linking affects search in 6.00x. The results are summarized in Table 4.5. In this table, in addition to the learning performance reported in Table 3.5, the result of the user study on the auto linking interface is also listed\(^\text{10}\). Furthermore, in Fig. 4-3 we also visualize the performance differences when the various interfaces were deployed. We present the improvement (i.e., time reduction in the upper panel and accuracy increase in the lower) from linking (red bars) and auto linking (black bars) as compared to null.

\(^{10}\)In this study, each facet of the experiment – except the deployed interface – was the same as that in Section 3.4.1. 1,000 HITs for the same 10 questions were collected on the Amazon Mechanical Turk. A between-subjects design was adopted. In this study we conducted the experiment with the three interfaces (i.e., null, linking, and auto linking) simultaneously. Additionally, the same quality control mechanism was employed.
Figure 4-2: The improvement in search time (upper panel) and accuracy (lower panel) with the linking (red bars) or auto linking (black bars) interface, with the null interface as the baseline. Learning performance improvement was measured in the Stat2.1x study. The 95% confidence intervals (shown as error bars) and significance test results (marked with red asterisk if the difference is statistically significant) are also provided.

In Table 4.5 and Fig. 4-3, we find that the auto linking interface also allowed learners to complete tasks with less time in most cohorts (except for subjects without a bachelor’s degree), as compared to the null interface. Search accuracy was also improved in the entire group of subjects, subjects with or without experience in Python, subjects with or without previous exposure to MOOCs, and subjects with a bachelor’s degree or higher. In addition, in comparing the improvements yielded by the linking and auto linking interfaces, we observe highly correlated and statistically significant improvements in mostly the same cohorts (except for the time reduction for subjects...
without a bachelor’s degree, the accuracy for subjects with Python experience, and the accuracy for subjects with previous MOOC exposure).

From the user study results discussed above, we observe that in most cases the auto linking interface was still able to help learners in the search task, but generally yielded slightly less improvement than the linking interface (except for the accuracy of the 6.00x study). The observation is interesting: our results in Section 4.4 show that some of the linking used in the auto linking interface were very different from those labeled by humans (e.g., the linking between videos and discussions in 6.00x); however, based on our user study experiment, learners seemed to benefit from both interfaces despite their using somewhat different linking annotations.
We believe the user study results support our previous conjecture that despite the few discrepancies between human- and machine-labeled linking, many of the differences could be attributable to the ambiguity of the underlying linking tasks. Therefore, both humans and machines created reasonable linking, and learners benefited from both the linking and auto linking interfaces. However, from the results we also believe that machines cannot match humans as to the depth of understanding of the learning content, and thus still make linking errors which confuse learners. Hence, usually smaller improvements in learning performance are measured when we replace human-labeled linking with the machine-generated ones. In Section 4.6, we continue this discussion about the difference between linking labeled by humans and machines, and investigate the difference patterns to explain why auto linking is still helpful. Below we investigate the other learning scenario: concept retention.

4.5.2 How automatic linking affects information memorization

Here we explore how the automatic linking affects learner performance in the concept retention scenario. Performance was measured by the numbers of unique key-terms in the learner-submitted essays. Table 4.6 compares the performance when an auto linking interface was deployed to results when linking or null was used (those reported in Table 3.6 for the Stat2.1x experiment). The user study for the auto linking interface followed the same experimental procedures described in Section 3.4.2, e.g., 1,000 HITs for the same 10 sampled topics on AMT, a between-subjects design, and a plagiarism check for quality control\textsuperscript{11}. Also visualized is the performance improvement (increased number of key-terms) from linking (red bars) and auto linking (black bars) as compared to null.

Similar to the results in the information search scenario, automatic linking also yielded performance improvements in the retention task. As compared to subjects assigned the null interface, learners who used auto linking mentioned more key-terms in their essays. The improvement is statistically significant in the entire group of.

\textsuperscript{11}This study was also conducted three months after the experiments on the null and linking interfaces were complete.
Table 4.6: Learner performance in the concept retention scenario in the Stat2.1x study. Performance was evaluated by the number of unique key-terms in submitted essays and measured within various cohorts using the null, linking, and auto linking interfaces.

<table>
<thead>
<tr>
<th>Number of unique key-terms</th>
<th>null</th>
<th>linking</th>
<th>auto linking</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall</td>
<td>4.39</td>
<td>4.91</td>
<td>4.83</td>
</tr>
<tr>
<td>Statistics</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>4.71</td>
<td>5.11</td>
<td>5.02</td>
</tr>
<tr>
<td>No</td>
<td>3.98</td>
<td>4.60</td>
<td>4.50</td>
</tr>
<tr>
<td>MOOCs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>4.83</td>
<td>5.14</td>
<td>5.07</td>
</tr>
<tr>
<td>No</td>
<td>4.27</td>
<td>4.77</td>
<td>4.75</td>
</tr>
<tr>
<td>≥Bachelor</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>4.73</td>
<td>5.23</td>
<td>5.04</td>
</tr>
<tr>
<td>No</td>
<td>3.98</td>
<td>4.60</td>
<td>4.46</td>
</tr>
</tbody>
</table>

experimental subjects as well as among the novice learners. Moreover, if we compare the improvement yielded by linking and auto linking, we find that the latter consistently showed a smaller increase in the number of key-terms.

The effect of automatic linking on the 6.00x retention task is also summarized in Table 4.7. The null and linking columns correspond to the results discussed in Table 3.7. The learning performance observed in the user study where the auto linking interface was deployed is listed in the auto linking column. Additionally, Fig. 4-5 visualizes the performance difference observed when different interfaces were used, and plots the increase in the number of key-terms from using linking (red bars) and auto linking (black bars) as compared to null.

From Fig. 4-5 and Table 4.7, we find that automatic and manual linking perform similarly. Both linking and auto linking interfaces equipped learners in each cohort to use more key-terms in their summaries of the assigned topics. However, at the 95% confidence interval, the improvement yielded from the auto linking interface was not statistically significant in the seven cohorts studied here.

Similar to what we found in the information search scenario, the user study results discussed here also show that despite the differences between manual and automatic links, learners benefitted from both. The results provide additional support for our

---

12In the study, the same experiment setup was adopted: 1,000 HITs for the same 10 topics and a between-subjects design. In addition, at the same time we conducted an experiment with the three interfaces (i.e., null, linking, and auto linking).
Figure 4-4: The improvement in the number of unique key-terms contained by submitted essays when the *linking* (red bars) or *auto linking* (black bars) interface was used, with the *null* interface as the baseline. Learning performance was measured in the Stat2.1x study. The 95% confidence intervals and significance test results are also provided.

conjecture that many disagreements between the manual and automatic linking come from the task ambiguity, and that the CRF model still makes reasonable decisions when linking learning objects. Hence, only a small degradation in learning performance was found when manual linking was replaced by automated linking.
Table 4.7: Learner performance in the concept retention scenario in the 6.00x study. Performance was evaluated by the number of unique key-terms in submitted essays and measured within various cohorts using the null, linking, and auto linking interfaces.

<table>
<thead>
<tr>
<th>Number of unique key-terms</th>
<th>null</th>
<th>linking</th>
<th>auto linking</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall</td>
<td>8.07</td>
<td>8.56</td>
<td>8.39</td>
</tr>
<tr>
<td>Python</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>8.64</td>
<td>9.09</td>
<td>8.74</td>
</tr>
<tr>
<td>No</td>
<td>7.64</td>
<td>8.20</td>
<td>8.14</td>
</tr>
<tr>
<td>MOOCs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>8.37</td>
<td>8.55</td>
<td>8.63</td>
</tr>
<tr>
<td>No</td>
<td>7.93</td>
<td>8.56</td>
<td>8.28</td>
</tr>
<tr>
<td>&gt;=Bachelor</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>8.60</td>
<td>9.13</td>
<td>8.77</td>
</tr>
<tr>
<td>No</td>
<td>7.21</td>
<td>7.91</td>
<td>7.88</td>
</tr>
</tbody>
</table>

Figure 4-5: The improvement in the number of key-terms contained in submitted essays when different interfaces were deployed in the 6.00x study. Also visualized is the improvement from using linking (red bars) and auto linking (black bars) as compared to null.
4.6 Difference pattern analysis

In the previous section, we provided evidence that automatic linking still benefit learning, and that the low similarity between manual and automatic linking causes only a small degradation in the observed learning performance. In this section, we attempt to provide explanations for why automatic linking are also helpful in learning – if not as much so as manual linking – by looking into the difference patterns between manual and automatic linking.

For the analysis, we choose the task of linking between video vignettes and discussion threads; this task had the lowest F1 score. We compare the linking predicted by our best automated system to human annotation. In the comparison, we first sampled 50 discussion threads from those threads which were linked to different vignettes in human and machine labeling. After reviewing the sampled threads, we identified four difference patterns:

1. Pattern 1: only annotators linked some vignettes to the thread.
2. Pattern 2: only machines linked some vignettes to the thread.
3. Pattern 3: both machines and annotators linked some but not the same vignettes to the thread; the non-overlapping vignettes belong to the same lecture video.
4. Pattern 4: both machines and annotators linked some but not the same vignettes to the thread; the non-overlapping vignettes belong to different lecture videos.

We categorize the sampled 50 threads into the four patterns in Table 4.8. Here we find that pattern 1 dominates in numbers. With this categorization, below we analyze each pattern and how it affects the learning experience in order to shed light on the user study results.

Pattern 1 includes 62% of the sampled threads. However, we believe that among the four, this pattern detracts the least from the learning experience. Because of the
Table 4.8: The number of threads categorized into the four difference patterns.

<table>
<thead>
<tr>
<th>Pattern 1</th>
<th>Pattern 2</th>
<th>Pattern 3</th>
<th>Pattern 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of threads</td>
<td>31</td>
<td>6</td>
<td>8</td>
</tr>
</tbody>
</table>

way we presented learning content and visualized linking, in this pattern, the auto linking interface simply regressed to the null; since the thread was not linked to any video vignette by machines, this thread was presented under a separate discussion tab. Although the regression increased the difficulty to access this thread, the user experience in interacting with the rest of the learning materials was almost identical. Thus learners still accessed the desired information from the rest of the materials to accomplish their tasks as they did in the interface driven by manual linking.

To illustrate this difference pattern and give a concrete example, we consider one specific thread from our set of 31. We present the content of this thread and its linked vignette in Fig. 4-6. In the left panel of the figure, we see that the discussion is about how dictionaries enable quick web searches in Google. In the right panel we observe that our TAs related this discussion to a vignette introducing the basic idea of a dictionary, while the machines linked nothing to this post. Thus, in our linking interface, when learners surveyed this vignette of dictionary introduction, this discussion about web search was rendered under the video, while in the auto linking interface no discussion was presented. However, the discussion only added fun facts and additional information to the vignette. Without the post, the concept of a dictionary could still be properly learned from the vignette. This example implies a negligible effect on learning for difference pattern 1.

In addition, we examined the relation between this difference pattern and task ambiguity, finding that in 19 of the 31 threads, one of the three annotators agreed with the machines (i.e., linking nothing to the thread). From this result we believe that many disagreements between machines and annotators derive from the conser-

---

13 Note that since a video vignette is defined as the video chunk aligned to one slide page, the content in a vignette can be well summarized by the aligned slide. For simplicity, here we show the aligned slide to represent the vignette.

14 Note that the manual annotation was obtained by taking the majority voting over the labeling of three TAs.
Random Question about Key:Value Concept

is this concept of dictionaries, or a key-value, used in a sophisticated way (complex keys that incorporate functions) by Google to enable quick web searches? Just trying to understand the power of dictionaries.

I think it's a lot different mechanism they use. Try reading up on this article on Wikipedia.

http://en.wikipedia.org/wiki/Google_search

Discussion thread

Linked vignette (Humans)

Linked vignette (Machine)

Figure 4-6: An example of difference pattern 1. The left panel shows a sampled discussion thread; the right panel presents the vignette linked to the thread by human annotators (upper right) and by the proposed CRF algorithm (lower right). The case where none of the vignettes is linked is represented by ∅.

Vativeness of the machines: the automated algorithm is inclined not to link videos and threads when the linking is ambiguous. This fact also supports our belief that this difference pattern has little negative effect on learning, since learners can focus on the case when the relation between learning objects is strong.

Six out of the 50 sampled threads were categorized into the second pattern. This pattern detracted from the learning experience and confused learners. In this case, the machines linked to discussion threads vignettes which were deemed irrelevant by annotators. Under these vignettes the discussion threads were shown in our auto linking interface. Learners were obliged to expend their cognitive capacity to understand these unrelated threads, and could have ended up feeling confused about why these threads were presented. Therefore with this type of difference pattern, learning performance was lowered.

We also give an example of difference pattern 2. In Fig. 4-7, we consider one specific thread from the set of six on the left; the content of the linked vignette labeled by humans and machines is presented on the right. The thread is meaningless discussion, but the machines linked it to a vignette describing a hash function. It is
def hashStr(s, tableSize = 101):
    number = ''
    for c in s:
        number = number + str(ord(c))
    index = int(number) % tableSize
    return index


Figure 4-7: An example of difference pattern 2. The left panel shows a sampled discussion thread; the right panel presents the vignette linked to the thread by human annotators (upper right) and the proposed algorithm (lower right).

obvious that this type of difference may distract or confuse learners, and thus detract from their learning experience.

We also examine the question of whether all disagreements result from algorithmic errors or whether some result from task ambiguity, finding that in the case of difference pattern 2, one of the three annotators agreed with the machine’s linking in 2 out of the 6 threads. Hence, we believe that not all differences in this category were the result of machine mistakes, and only a portion of the differences caused declines in learner performance.

As for the third pattern, 8 out of the 50 sampled threads were classified in this category. Similar to the first pattern, we also believe this difference pattern had little negative effect on learners given our interface design. In this pattern, although the machines and humans linked different vignettes to a thread, these vignettes belonged to the same lecture video. With our interface design, this thread was presented under the same video but was merely aligned to different parts of the video scrubber. Therefore, this difference pattern did not greatly change the user experience.15

An example of difference pattern 3 is shown in Fig. 4-8. As we can see, the thread

---

15In 4 out of the 8 threads, one of the three annotators linked the same video vignettes as the machine. This also supports our claim that the proposed automated system links reasonable vignettes; thus it is very likely the disagreement did not affect learners negatively.
why memf(*x)

I think memf() is enough.

It is a question of building a more flexible widget. Maximum bang for your buck.
memf("args", "keyargs")
would be even more flexible but create more "why?" questions.
Oh got it. Because if * we can pass function with multiple arguments
But in our case, i.e. Fibonacci, it is not required. Right???

Discussion thread

Function arguments as a sequence

1) Capturing arguments of a call:

def f(*x):
    """tuple of args"
    x = (1, 2, 3)

2) Using sequence to supply arguments:

result = f(*x)

Linked vignette (Humans)

"Memoization"

if works for functions with hashable (immutable) arguments:

# a simple wrapper function: memoize(f) = memoize(f())
def memoize(f):
    cache = {} # list or dict?
    def memf(*x):
        if x not in memo.cache:
            memo.cache[x] = f(*x)
        return memo.cache[x]
    return memf
    memo.cache = {} # all code...
    return memo

Linked vignette (Machine)

Figure 4-8: An example of difference pattern 3. In this example, the thread (left panel) was linked by humans and machine to two vignettes (right panel) in the same lecture video. The two vignettes are closely related to each other, and the difference in presenting these two ways of linking was only minor.

was relevant to both vignettes linked by humans and machine. When the two various ways of linking were presented in our interface, qualitatively the difference was minor, which supports our previous claim.

A total of 10% (5 out of 50) of the sampled threads were categorized as the last pattern. As with pattern 2, this difference category confused the learner, detracting from the learning experience, since in auto linking the discussion threads were linked to lecture videos which were totally different from those labeled by humans; here the proposed automated linking algorithm agreed with any of the three annotators in only 1 out of the 5 threads. Thus, we conclude that this difference pattern is less likely to be caused by task ambiguity; it is very likely that videos that were linked only in the automated system were irrelevant to the threads.
Finding recurMul() in symbol tables

Hi. In the slides for this video segment, Professor Gomson seems to suggest that each call to recurMul looks up its procedure binding in the global environment, no matter where that call is on the stack (notice that the environment pointers all point to the global environment in the slides). Have I understood that correctly? If so, that seems very different behaviour from regular variables (binding principles), since if I attempt to refer to a variable not in the current frame, Python throws an exception.

Many thanks in advance for your help.

In "all" cases, any variable will be looked up in the global environment if not found in the local one. This is why the language is actually usable (since pretty much everything is a variable of some form, and most of them live in the global scope).

Discussion thread

Let's try it out

```python
def recurMul(a, b):
    if b == 1:
        return a
    else:
        return a + recurMul(a, b-1)

recurMul(2, 3)
```

Linked vignette (Humans)

Global variables

- Use with care!!
- Destroy locality of code
- Since can be modified or read in a wide range of places, can be easy to break locality and introduce bugs!!

Linked vignette (Machine)

Fig. 4-9 shows an example of difference pattern 4. In the posts here, learners discuss how the Python interpreter utilizes symbol tables to keep track of variable bindings in recursion, which was explained exactly in the vignette linked by our TAs. In contrast, the proposed CRF algorithm linked this thread to a less relevant vignette belonging to another lecture video describing global variables. Learners could have been distracted by the discussion of the symbol tables when viewing this video.

From this analysis we can find that although the similarity between automated and manual linking was low in some tasks, we find reasonable many of the differences resulting from task ambiguity and the linking labeled by both machine and humans; otherwise the differences were properly presented in our interface. Such differences were recovered or ignored easily by learners and thus had little effect on user experience. Therefore, we also observe considerable improvement in learner performance when the auto linking interface was deployed. Note that this analysis was performed for the process linking video vignettes and discussion threads. We believe that this
conclusion generalizes to the linking of other materials, since similar automated algorithms were utilized.

4.7 Comparison with the edx interface

In the above studies, we have shown that when we present learning content to learners, if we are able to visualize the relations among content, learners achieve better performance in completing the assigned learning tasks. Furthermore, we demonstrated that we can obtain the relations either from human annotators or the proposed CRF linking algorithm. These studies were conducted by comparing the linking and auto linking interfaces to the null interface, a baseline that implements the conventional strategy for delivering learning materials online. In this section, we investigate whether our linking framework provides added value to the interface currently deployed for MOOC platforms (here we choose as our baseline the edX website).

4.7.1 The edx interface

To ensure a consistent deployment for the user study, rather than using the edX website directly, we implemented our own edx interface to conduct the AMT study. A screenshot of this interface is presented in Fig. 4-10. We reproduced the design of the interface and the content layout from the edX website in order to offer learners a user experience identical to the one they use when engaged in a state-of-the-art MOOC platform. The only difference between our edx interface and the real platform is that here, we additionally provided the search mechanism for accessing course materials; this was also done in order to ensure a comparison consistent with our linking, auto linking, and null interfaces.

In this interface, instructors upload a deck of lecture slides beneath the paired lecture video. The associated slides are presented as a link under the video player for learners to refer to. On the edX website, to motivate learners to engage in discussion and to organize the voluminous forum postings, learners were allowed to post under a lecture video to specify the relation between their discussions and the lecture; these
Figure 4-10: The implemented edx interface that reproduced the design and layout of the edX website to offer learners a user experience similar to that of a real MOOC, except that we added the additional search tool to access course materials. This interface was used to investigate how much added value our linking framework provides for state-of-the-art MOOC platforms.

posts were directly rendered under the video for future learners. This functionality was also implemented in our edx interface.

From the design of the edx interface we find that this interface can be interpreted as another approach to educational content linking on different information levels: lecture slides were linked on the lecture level rather than the page level as in the linking interface; the relation between discussions and lecture videos was inferred from learner choices rather than being based on the content of the material; the textbook was still presented separately. Thus, as compared to the null interface, the edx interface serves as a baseline in the comparative study for a different purpose. We implemented the edx interface in order to investigate how much added value our
linking framework provides for state-of-the-art MOOC platforms. In contrast, we utilized the null interface to explore the fundamental research question asked in this thesis: does linking help learning?.

4.7.2 User study, results, and discussions

To examine what value the proposed linking framework adds to state-of-the-art MOOC platforms, we conducted another user study. In this study, again we published 1,000 HITs on AMT for each of the two learning scenarios, and learners in this study were to use the edx interface to accomplish their tasks. By comparing the learning performance measured here to the results from the null, linking, and auto linking interfaces, we can investigate whether our educational content linking framework improves on the current MOOC design. Note that in this study, except for the deployed interface for completing tasks, the remaining experimental setup was identical to the other user studies discussed in this thesis (e.g., the same 10 sampled questions and topics, the same number of rewards, the same quality control mechanism, and the between-subjects design). Furthermore, for simplicity in this chapter we only investigated 6.00x for our study, and the user study was conducted together with the other three (i.e., null, linking, and auto linking) interfaces.

Table 4.9 summarizes learner performance (evaluated by the average search time and accuracy) in the information search scenario. Columns 1 to 3 and 5 to 7 correspond to the performance measured when the null, linking, and auto linking interfaces are utilized. These results are identical to those reported in Table 4.5. The user study result when the edx interface was deployed is listed in columns 4 and 8. Here, we employ the same dividing criteria to stratify learner backgrounds (i.e., prior knowledge, experience in MOOCs, and highest degree).

Since our goal is to examine how much added value our educational content linking framework brings us, in Fig. 4-11 we again visualize the differences in performance yielded by different pairs of interfaces. In this figure we present the improvement (i.e., time reduction in the upper panel and accuracy increase in the lower) from linking (red bars) and auto linking (black bars) as compared to null, as well as the
Table 4.9: Learner performance in the information search scenario in the 6.00x study. In addition to the results reported in Table 4.5, we list performance (evaluated by the average search time and average accuracy) measured when the *edx* interface was used.

<table>
<thead>
<tr>
<th></th>
<th>Average search time (seconds)</th>
<th>Average accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><em>null linking</em></td>
<td><em>auto linking</em></td>
</tr>
<tr>
<td>Overall</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>443</td>
<td>349</td>
</tr>
<tr>
<td>Python</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>419</td>
<td>323</td>
</tr>
<tr>
<td>No</td>
<td>463</td>
<td>378</td>
</tr>
<tr>
<td>MOOCs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>427</td>
<td>336</td>
</tr>
<tr>
<td>No</td>
<td>454</td>
<td>357</td>
</tr>
<tr>
<td>≥Bachelor</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>472</td>
<td>359</td>
</tr>
<tr>
<td>No</td>
<td>399</td>
<td>331</td>
</tr>
</tbody>
</table>

Table 4.10: Learner performance in the concept retention scenario in the 6.00x study. In addition to the results reported in Table 4.7, we list the number of unique key-terms in submitted essays measured when the *edx* interface was used.

<table>
<thead>
<tr>
<th></th>
<th>Number of unique key-terms</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><em>null linking</em></td>
</tr>
<tr>
<td>Overall</td>
<td>8.07</td>
</tr>
<tr>
<td>Python</td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>8.64</td>
</tr>
<tr>
<td>No</td>
<td>7.64</td>
</tr>
<tr>
<td>MOOCs</td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>8.37</td>
</tr>
<tr>
<td>No</td>
<td>7.93</td>
</tr>
<tr>
<td>≥Bachelor</td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>8.60</td>
</tr>
<tr>
<td>No</td>
<td>7.21</td>
</tr>
</tbody>
</table>

improvement from *linking* (blue bars) and *auto linking* (orange bars) as compared to *edx*. By looking at the blue and orange bars, we understand the potential of our linking framework to improve content delivery in current MOOC platforms. The red and black bars are plotted here for comparison to our previous results; they are identical to those in Fig. 4-3.

In Table 4.10 we report the observed performance in the concept retention scenario. In addition to the results discussed in Table 4.7, which are listed in columns 1 to 3 here, in column 4 we present the performance measured for learners who used the *edx* interface for their tasks. Additionally, to visualize the added value yielded by the proposed linking framework, we also plot in Fig. 4-12 the differences in the number of key-terms when the various interfaces were deployed.
Figure 4-11: Improvement in search time and accuracy for different interfaces. Plotted are the improvement from using linking (red bars) and auto linking (black bars) as compared to null, and the improvement from linking (blue bars) and auto linking (orange bars) as compared to edx.

From the linking – edx and auto linking – edx bars (i.e., the blue and orange bars) in Fig. 4-11, we find that the proposed linking interfaces (driven by either manual or automatic links) allowed learners to find content more quickly but with similar accuracy. Comparing each linking – edx and auto linking – edx bar to the corresponding linking – null (i.e., red) and auto linking – null (i.e., black) one, less time reduction is observed in general, and the reduction is significant in fewer groups of subjects. For the linking – edx and auto linking – edx bars of subjects with experience in Python, subjects with previous exposure to MOOCs, and subjects with a bachelor’s or higher degree, as well as the auto linking – edx bar of subjects
Figure 4-12: The improvement in the number of unique key-terms contained by submitted essays when different interfaces were deployed. The bars are pictured as in Fig. 4-11.

without previous exposure to MOOCs, the differences are not statistically significant. However, their counterparts in the linking – null and auto linking – null bars are significant. Results here suggest the added value that the proposed linking and auto linking interfaces provide for current MOOC platforms. Furthermore, in the search accuracy, statistically significant improvement was shown only for the auto linking – edx bars of the entire group of subjects, subjects with experience in Python, and subjects without previous exposure to MOOCs. As compared to the linking – null and auto linking – null bars, statistically significant difference is observed in a fewer number of cohorts. These observations suggest that the edx interface is a better baseline (in terms of yielding better learning performance) than the null one. The reason is self-evident: the edx interface implements its own linking, which have been shown to be able to make course materials more accessible and improve learning.

The proposed linking interfaces also yielded more key-terms consistently over each cohort as compared to the interface reproducing the current MOOC design (Fig. 4-12). Furthermore, in this learning scenario, the null and edx interfaces seemed to perform similarly, and therefore the difference between the linking – edx and auto linking – edx bars when compared to the linking – null and auto linking – null ones in the same cohorts is much less obvious than that in the information search
scenario. We surmise that this difference results from the nature of the two scenarios. In comparing the null and edx interfaces, one of the most significant changes is that in the edx interface relevant discussions were stacked beneath the lecture video. Discussions are typically initiated because of confusion about specific problems or concepts in the learning content; they contain useful information to solve questions asked in the search scenario. However, information in these posts is fractional and thus it is challenging to learn about a topic systematically from these posts. Therefore, the improved navigation over discussions was much more helpful in our search tasks.

Our results show that the proposed educational content linking framework potentially allows learners to find desired learning content more quickly than the current MOOC interface design with comparable accuracy; with the proposed interfaces, subjects also retained more information after learning with the assigned topic in the same amount of time. Since the edx interface also partially implemented linking, we surmise that the improvement resulted from several differences in the interface design. First, the page-level alignment between slides and lecture videos better reflects the structure and emphasizes the sub-goals of videos. Second, the relation between discussions and videos tagged by learners can be noisy and distract other learners; taking into account the topical relevance between content contained in the two types of materials helped to maintain the tagging quality. Third, some discussions might be related to only parts of the videos. This information is not available in the current MOOC platforms. Fourth, it might be worthwhile to visualize the relations between lecture videos and external resources, such as a recommended textbook. We believe that by integrating these features properly into the current platform design, learning experience in MOOCs would be enhanced significantly.

4.8 Conclusions

In this chapter we investigated our second research question: can linking be done at scale? For this question, we formulated the linking annotation as a sequential tagging problem, and proposed an automatic content linking algorithm based on
CRFs. In the algorithm, to infer the linking, a variety of features were utilized: lexical similarity, transition, visual, and metadata features. In analyzing the difference patterns between automated and manual linking, we found that many differences resulted from task ambiguity and had little negative effect on learners. Hence, in our user research we observed similar improvement in learning performance when the auto linking interface replaced the linking one. We concluded that our linking framework can be realized at scale with an automated algorithm.

Furthermore, we also compared our linking and auto linking interfaces to a reproduction of the edX website, and explored how these interfaces support learners in completing assigned tasks. In the user study we also demonstrated that, as compared to the current design in MOOC platforms, our educational content linking framework still helped learners find information more quickly and retain more concepts. We believe that the proposed linking framework enriched the current MOOC interface design. We envision engaging learners with more accessible course materials and a better learning experience powered by content linking.
Chapter 5

Conclusions

This dissertation introduced educational content linking: a framework for organizing course materials to make content more accessible for learners. In this chapter we conclude by summarizing the main contributions of this thesis and discuss possible directions for future work.

5.1 Summary and contributions

This thesis contributes to the research community by proposing a framework for educational content linking. This framework provides better navigation over learning materials and improves the learning experience. Around the framework we conducted two lines of studies to answer two research questions: 1) Would it help learners if we were able to link course materials using human annotators? and 2) Can the courseware be linked at scale using machine learning methods?

In exploring the first question, this thesis makes three main contributions: the linking annotation, the interface design, and the evaluation.

- **The linking annotation.** We provide a definition of linking and a formulation of labeling relations among learning materials as an alignment and binary classification problem. We design the workflow of annotating linking with researchers or with collaboration between course staff and online workers. Linking is an abstract concept; this contribution makes it concrete.
• **The interface design.** We design an interface to deliver learning content with the visualization of linking among the materials. The interface can provide pedagogical benefits through improved content navigation.

• **The evaluation.** We conduct a large-scale user study with online workers and two learning scenarios to investigate specific learning mechanisms: search and retention. We argue that this study can measure the benefits of pedagogical intervention reliably with reasonable cost, when the underlying learning goals in the study are clear to participants. The study result shows that the proposed linking framework indeed improve learning outcomes in the investigated search and retention scenarios.

In the second part of this thesis, we investigated the second research question, and offer two main contributions: the automated linking algorithm and the comparison to a currently deployed MOOC interface.

• **The automated linking algorithm.** We propose an automated linking algorithm based on CRFs and multimodal features. In our large-scale user study we demonstrate that, despite the differences between the manually and automatically generated linking, most differences can be properly presented in our interface or easily ignored by learners, and thus the interface powered by automated linking can still lead to better learning performance than the unlinked interface. This result suggests that the proposed linking framework can be realized at scale with an automatic algorithm based on machine learning techniques.

• **The comparison to a currently deployed MOOC interface.** In addition to the conventional unlinked content delivery, we explore the added value our linking framework can potentially provide to a currently-used MOOC interface. The user research result suggests that the framework proposed here can possibly enrich the design of the studied MOOC interface, engage learners with more accessible learning content, and improve learning outcomes.
5.2 Future work

This dissertation showed the potential of educational content linking in engaging learners with a better learning experience. This conceptual idea can be further verified, refined, and applied to various circumstances to improve learning.

5.2.1 Learning platforms of the future

Our user research results demonstrate several possible ways in which linking can offer learners a better experience. We envision that the future learning platforms will engage learners with more accessible learning content. Although in this thesis we investigate linking in two learning tasks, search and retention, there are many other aspects of learning that could potentially benefit from the proposed framework. For instance, when solving the problem sets or performing online lab experiments, with better organized learning materials, confused learners would be more likely to receive proper assistance from the content. Furthermore, our studies were conducted with online workers. Research implementing the proposed experimental pipeline in a MOOC environment would be valuable to clarify the mechanism by which linking helps actual learners.

Instead of reproducing our entire implementation of linking, separate components in our pipeline also inspire directions of design for future platforms. For example, the automated linking algorithm could be applied to filter noisy posts and improve the quality of discussions; instructors could utilize the algorithm to discover relevant learning content to enrich or reorganize their lectures. Our presentation of lecture videos provides design implications for better video interaction with visualized structure and subgoals. The design of our linking interface suggests an elegant way to offer recommended readings. These components lead to new avenues to improved MOOC platforms.
5.2.2 Towards a variety of course subjects and material types

This thesis focuses on two STEM courses: statistics and the Python programming language. However, as we mentioned in Chapter 1, there have been over 4,000 MOOCs on the Web with subject fields ranging from science and engineering to humanities and law. In addition to their topics, these MOOCs span various applied pedagogies, course designs, and methods for content organization or delivery. Several interesting questions that should be explored are "Whether our linking framework can be applied to other subject fields", "which fields, pedagogies, designs, and content organization can benefit from linking", and "how various conditions interact with the idea of linking". We believe a wider deployment of this framework in various MOOCs can elucidate these questions.

With a wider deployment, the involved types of course materials would also vary. For example, some MOOCs heavily emphasize problem sets, while others stress online labs. For the proposed framework to be of more general use, we also must answer questions such as "how to extend our implementation, from linking annotation to visualization, to accommodate these variations" and "whether and how the conclusions made in this dissertation are affected by the use of different types of materials". Our initial foray of adding forums to our implementation is a good illustration of how to investigate these questions.

5.2.3 More sophisticated algorithm for linking at scale

In Chapter 4, we demonstrated that many differences between manual and automated linking have little negative effect on learners. However, there is still a considerable portion of disagreement which can confuse learners. We believe that a more sophisticated machine learning model would yield a deeper and more comprehensive understanding of the course materials, and thus generate linking which is more similar to the human annotated one.

One promising model is the attention-based neural network. In this model, first proposed for machine translation [6], for each word in the sentence of the target
language, the network learns a weight for each word in the source sentence. The weight represents how relevant the source word is in predicting the target word. When applying this model to our linking problem, the target tokens to be predicted would be the linking configuration (e.g., the linked slide index, or whether two objects are linked), and the source tokens would be two sequences of learning objects to be linked. This network would focus its attention on informative learning content when deciding the linking configuration; this mechanism is similar to how humans generate the linking.

More informative features would also help the model to yield better linking results. In addition to the information extracted from learning content, such as the lexical and visual features utilized in our method, user behavior is another excellent resource for predicting linking. In this thesis, we demonstrate the usage of learner-generated tags about discussion posts in the linking algorithm. Aside from these tags, click logs and browsing histories are very likely to aid in linking inference. Kim shows that the aggregation of learner video interactions can reveal the underlying video structure and provide implication for video authoring and interface design [63]. We believe that by grouping the browsing history and summarizing clickstream patterns, we could better understand the relevance among learning materials and extract informative features for linking prediction.

Beyond these machine learning techniques, crowdsourcing (or learnersourcing) is an alternative for linking at scale. Li and Mitros proposed a learning module where learners can collaboratively recommend additional learning objects and manage the recommended materials for future learners [79]. We envision a linking system which allows learners and machines to author, edit, and manage the linking of course materials in a collaborative way.

Furthermore, portability is another issue that should be investigated towards a scalable linking system. In this thesis, we adopted a 5-fold cross validation technique to obtain training and testing sets from the same MOOC. This experimental setup raises the question of why we need the automated linking, since the manual labeling of the entire MOOC is available. Hence, more realistic conditions should be explored,
where for instance the automated algorithm is trained and tested on the same course subjects but different offerings, or even on different MOOCs.
Appendix A

Sampled problems and topics for the user study

A.1 Sampled problems

A.1.1 Problems of Stat2.1x

The following problems were used in the information search scenario of the Stat2.1x user study.

Q1 The table below shows the distribution of the ages of people who died by gunfire in the U.S. during one week. Based on the table, how to compute the height of each bar in the histogram.

<table>
<thead>
<tr>
<th>age (years)</th>
<th>percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>15-25</td>
<td>44</td>
</tr>
<tr>
<td>25-35</td>
<td>20</td>
</tr>
<tr>
<td>35-55</td>
<td>16</td>
</tr>
<tr>
<td>55-85</td>
<td>20</td>
</tr>
</tbody>
</table>

Q2 If both X and Y axis have the same unit (e.g., cm, degree, pound, ...), do I have to convert them into standard unit (i.e., z-score) in order to calculate correlation coefficient (r)?
Q3 The average height of a large group of children is 43 inches and the SD is 1.2 inches. The average weight of these children is 40 pounds and the SD is 2 pounds. The correlation coefficient (r) between the two variables (height and weight) is 0.65. What is the equation of regression line for the two variables?

Q4 The average height of a large group of children is 43 inches and the SD is 1.2 inches. The average weight of these children is 40 pounds and the SD is 2 pounds. The correlation coefficient (r) between the two variables is 0.65. How to estimate the height of a person whose weight is 44 pounds with the equation of regression line?

Q5 What is the formal definition for \(X^{th}\) percentile, where \(X\) is a general, real number between 0 and 100?

Q6 When we compute the residual, the error is the distance between an actual point and the regression line. Should we use the vertical distance (i.e., draw a vertical line from the actual point to the regression line, and take the distance between the actual point and the intersecting point), perpendicular distance (i.e., from the actual point, draw a line which is perpendicular to the regression line), or the horizontal distance?

Q7 What is the range of value of correlation coefficient (r)? Can it be 5, negative number, etc.

Q8 What is the definition of a football-shaped scatter plot? How does a football-shaped scatter plot look like?

Q9 In a stem and leaf plot, we have elements something like:

16 | 1234
17 | 89
18 | 56

What exactly does this mean?

Q10 In the following, we show the distribution of midterm scores in a statistic class. Please find the 'inter-quartile range'.

A.1.2 Problems of 6.00x

The following problems were used in the information search scenario of the 6.00x user study.

Q1 x is a tuple and $x = ('John', 'Hello', 'A', 'Hi$). What is the value of $x[2]$?

Q2 What error (if any) is raised when the following code snippets are attempted?

```python
mylist = [10, 20, 30]
mylist.index(11)
```

A: ValueError
B: TypeError
C: SyntaxError
D: NameError
E: No error is raised

Q3 What method is called when an object is created?

A: self
B: obj.self
C: init
D: __init__
E: new

Q4 True or False?

- A Python class is an example of data abstraction.
Q5 A dictionary is an immutable object because its keys are immutable.
   A: True
   B: False because its keys can be mutable
   C: False because a dictionary is mutable

Q6 True or False?
   - Declarative knowledge refers to statements of fact and imperative knowledge
     refers to 'how to' methods.

Q7 True or False?
   - Every problem which is computable can be computed with a set of six
     primitive operations.

Q8 For the following explanation of type of programmatic model, fill in the blank
with the appropriate model the definition describes.

   A __________ model is one in which randomness is present, and variable states
   are not described by unique values, but rather by probability distributions.
   The behavior of this model cannot be entirely predicted.

   A: continuous
   B: deterministic
   C: discrete
   D: dynamic
   E: static
   F: stochastic

Q9 x is a list and x = [1, 4, 3, 0]. Specify the value of x after executing the
following expression:

   » x.append(7)

Q10 Samples were taken from a distribution, and the histogram of those samples
    is shown here:
Which of the following distributions were the samples taken from?

A: Uniform Distribution
B: Exponential Distribution
C: Normal Distribution

A.2 Sampled topics

A.2.1 Topics of Stat2.1x

The following topics were used in the concept retention scenario of the Stat2.1x user study.

1. Regression
2. Residual
3. Normal distribution
4. Percentile
5. Histogram
6. Standard deviation
7. Mean
8. Scatter plot
9. Median
10. Correlation

A.2.2 Topics of 6.00x

The following topics were used in the concept retention scenario of the 6.00x user study.

1. Operation
2. Function
3. Computation complexity
4. Dynamic programming
5. Graph
6. Object
7. Iteration
8. Class
9. Recursion
10. Sort
Bibliography


