From Concepts to Curricula: Content Exploration, Generation, and Evaluation

Fatima Al-Raisi

Thesis Commitee:

Jaime Carbonell, Chair	(Carnegie Mellon University)
Chinmay Kulkarni	(Carnegie Mellon University)
Jordan Rodu	(University of Virginia)
Noah Smith	(University of Washington)

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

Language Technologies Institute
School of Computer Science
Carnegie Mellon University

November 2017

Contents

List of Figures

List of Tables

Abstract

While technological progress made more content readily available to learners, the information load makes it hard to navigate the content space and attain learning goals. The exponential growth of specialized knowledge challenges learners, scholars, teachers, and educational institutions. For novice learners, knowing where to start and what courses contribute towards educational goals, and in what sequence, are key problems. For scholars and researchers, keeping abreast of latest developments in their field, embarking on new or interdisciplinary research, or gaining deeper understanding of the foundations of a field, all require understanding of how the knowledge space is structured and connected. For educators and educational institutions, developing curricula of proper coverage and design implies knowledge of content mapping, overlap, and dependency. For emerging online learning organizations, course recommendation and personalization of the learning experience require complex learner-content modeling. As the space of knowledge grows in scope and connections, new questions arise: How do we define coherent "modules" or "topics" in this ever growing space? How do we represent content at different levels of abstraction? Can we discover latent patterns and relations in the content space? What are the characteristics of well-designed curricula? Can we formalize and guide the personalization of learning paths? Can we automate curriculum planning and generation? What is common between all these questions is the their relevance to *content*. In this thesis, we propose to model content at the **concept**, **module**, and **curriculum** levels which we define in detail in this proposal. At each of these levels, we focus on three main problems: **representation**, (planning and) **generation**, and **evaluation**. We propose models, methods, and evaluation criteria and metrics for the specific tasks we define to tackle these main problems. The representation, data structures, results and resources at each level serve to solve problems in the next level. Ultimately, the goal is to fully automate the navigation of the knowledge space given specific educational goals (coverage) and constraints defined on the learner's resources.

1 Introduction

This thesis aims to model different aspects of content including conceptual overlap and conceptual dependencies between different units of text bearing meaning and function. These relations are studied at different levels from basic concepts to sequences of topics or subjects. The goal is to learn optimal content structuring in terms of coherence, cohesion, and suitability for specific learning goal. At the core of this work is modeling and learning dependency relations between documents and dependencies among concepts within and across documents. For examples, if news article j refers to an event or an entity described in more detail in news article i, then j *depends* on (the information in) i. Similarly, if course j builds on concepts explained in course i, then course j *depends* on course i or, equivalently, i is a prerequisite for j. In mathematics, the concept of "permutations" depends on "factorials" and "factorials" depend on "multiplication." Knowing dependency relations among concepts and between documents is important for content structuring, coherent content generation, and presentation/ordering of results in various information retrieval tasks.

We frame this problem and contextualize it in the educational domain due to its resemblance to curriculum planning and generation problems in the education domain (1) and the availability of training data in the form of document dependencies defined as prerequisite relations between courses. The larger scope of this project in the educational domain includes automatic curriculum planning and generation. To produce a coherent curriculum, we need to know dependency relations between concepts within a course and proper ordering of courses in a sequence of courses.

The space of specialized knowledge continues to grow larger and denser. Efficient navigation of this space becomes a challenge facing learners, scholars, instructional designers, and educational institutions. Moreover, it is argued that the information load is leading to other unfavorable outcomes such as overspecialization, increased dependency among team members [20], and gaps in foundational knowledge for highly specialized individuals. These are linked to the essential problem of difficulty navigating the learning space and prioritizing learning objectives in terms of which content to cover at different stages of learning. We propose to focus on the problem of navigating the content space and creating learning paths that meet specific learning goals and are conditioned to the individual learner's background, preferences and constraints. To do so, one must first have a proper representation of the content and its structure and then apply search methods to exploit this structure, answer relevant questions, and generate desired output that can be tested and evaluated. More specifically, we propose to build tools and theoretical models to:

1. **Represent** educational data at different levels of abstraction for efficient computational processing and exploration.
2. **Analyze and Evaluate** educational data by detecting patterns of overlap, coherence, conceptual dependency, and temporal dynamics of content drift at different levels.
3. **Generate** educational knowledge in the form of modules, courses and curricula from the collected repository of educational data given a specific set of educational requirements in terms of intended learning outcomes and constraints defined in terms of consistency, coverage, and optimal coherence-overlap-redundancy tradeoff.

2 An Overview of Educational Data Mining

Educational Data Mining (EDM) [33] is an emerging field concerned with the application of computer-based pattern detection methods for the purpose of collecting, processing,

and analyzing educational data and ultimately discovering knowledge useful for the advancement of education. A related field is "Learning Analytics" (LA) which is defined as "the measurement, collection, analysis and reporting of data about learners and their contexts, for purposes of understanding and optimizing learning and environments in which it occurs [23]." The "Big Data" era in which large amounts of educational data have become available onlinewitnessed the emergence and growth of this line of work. These trending research areas have tackled different types of problems in the education domain. They can be broadly classified into of the following themes:

- Modeling learners: this includes work on identification of different types of learners and learner communities, work on modeling students progress and factors influencing students retention and attrition.
- Modeling learning media/paradigm: this includes work on different learning media and learning paradigms such as online-learning, MOOCs, synchronous and asynchronous learning, and the design and application of different educational technologies for teaching and assessment.
- Communication in learning: this includes work on modeling the communication in different learning settings and detecting patterns in the interaction styles among students and educators and its relationship with success rates and effective learning.
- Learning management and support: this includes work on developing learning management systems (LMS) and various educational technologies for the organization and delivery of content, and assessment.

The vast majority of research in EDM and LA has been focused on modeling the learner, the technology, and the pedagogy dimensions. Authors in [18, 19] argue that there are four main dimensions in learning: the learner (or learning style), content, technology, and pedagogy.

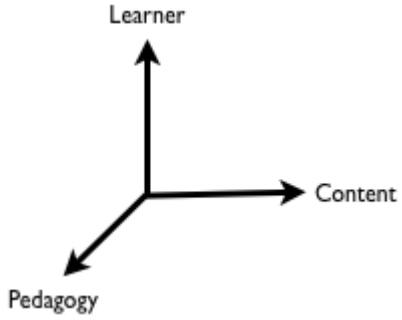

Figure 1: Three main dimensions in learning

3 Contextualization and Contribution

We propose to focus on the "content" dimension in this work and the various relations between units of content. While technological progress made more content readily available to a wide range of audiences, the information load poses new challenges for learners. We use the term "learner" generically. A learner can be a beginner trying to make use of the content and certification opportunities available online, a student already enrolled in a degree program and searching for specific information to fill gaps in his/her knowledge, or a scholar exploring new research areas or trying to gain insights into relevant results and multiple findings across fields. All of the above require knowledge of structure and relations in the content space and a way of effectively and efficiently navigating the large

space of specialized knowlege. We consider three levels of educational content: concepts, modules (or courses), and curricula (which we also refer to as *programs*). For the concept level, the focus is on representation and organization. For courses and programs, two additional problems are studied: analysis and generation. The specific problems studied in representation, analysis, evaluation, and generation are detailed in later sections.

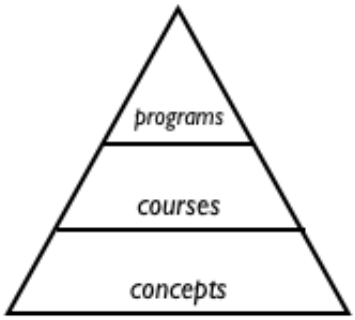

Figure 2: Levels of educational content: concepts, courses, and programs

Relative to the size and rapid growth of research in educational data mining [32, 29, 38, 6, 39], little research has been dedicated to modeling educational *content*. However, the availability of course textual description through MOOCs and websites of universities has facilitated the emergence of research in this area. Automatic course prerequisite prediction based on course textual content is a relatively new problem. This problem is also of great relevance in the MOOCs and open-access-to-information era. A learner navigating this vast and rapidly growing space needs to know cross-institutional dependencies and overlap between content from different sources in order to navigate the space efficiently and effectively. In recent years, a few research studies focusing on the content aspect have emerged [46, 3, 41, 31]. In [41], the authors try to find and compare the coverage of computer science curricula using text mining and analytics methods. In [3], the goal is to generate study plans from a "noisy" concept graph constructed from Wikipedia. The model in [3] classifies relations between concepts into one of three types: dependent, co-dependent or no relation, using graph-based features. More recently, predicting prerequisite relations between fine-grained concepts is formulated as a binary classification problem in [31] using heavily engineered contextual and structural features and applying baseline classifiers such as Naïve Bayes, SVMs, and logistic regression.
Another related line of work is on adaptive instructional policies that attempt to optimize student performance and uses student performance trajectories as learning data [5, 12, 37]. The models are based on Bayseian Optimization such as multi-armed bandits [5], Partially Observable Markov Decision Process [37], and Bayesian Knowledge Tracing [7, 48] in combination with reinforcement learning. These approaches differ in that they use an explicit state-space model of knowledge and in using learning data (student scores in assessment) to guide optimization. They assume prerequisite constraints as input which is the objective of learning in this thesis. We choose not to optimize on student performance metrics for a number of reasons including maintaining generalizability of the method to learning dependencies between documents and features in any domain. Another main difference between the above line of research and this thesis is our primary focus on the content dimension, independently of student assessment data as we optimize on generic content metrics. Third, we use both supervised and unsupervised or semi-supervised methods that do not require assessment data. Fourth, we consider content at a larger

scope and higher levels of abstractions; i.e., we do not focus on a single educational concept or topic but a network of interrelated concepts that form units of study and larger curricula.

Perhaps the most relevant work to this problem is the work on concept graph learning from educational data [46] where the task is to automatically learn cross-institutional prerequisite relations using training data consisting of observed prerequisite relations between courses from the same institution and a concept representation of each course. The goal is to map the courses, using their concept representation, into a canonical space of concepts and do the reasoning in that space. The concept graph is used in this case as an intermediate step or interlingua in a transfer learning setting where the inference is mapped from the course space to the concept space and then projected back to the course space to predict missing links. Figure 3 illustrates this process.

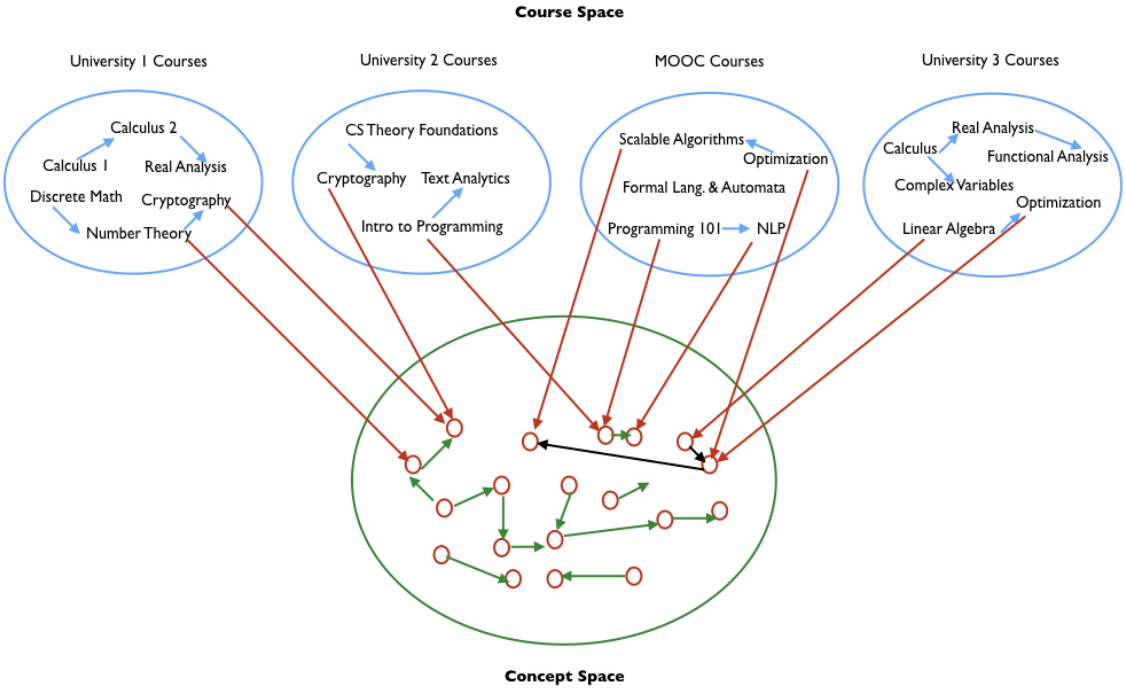

Figure 3: Transferring edge prediction from course space to concept space and projecting back to the course space

For example, the figure shows that the dependency between linear algebra and scalable algorithms is not directly observed in the course space, but can be inferred from the mapping in the concept space.

In [46], course prerequisite learning is defined as an edge prediction problem, referred to as CGL henceforth. The problem is first formulated as a binary classification problem where the edge between two new courses x_i and x_j is classified as either present or not.

The formulation for the classification approach is:

$$\min_{A} \sum_{i,j} \left(1 - y_{ij}(x_i^T A x_j) \right)_+^2 + \lambda \|A\|_F^2 \tag{1}$$

where n is the number of courses in a training set,
p is the dimension of the concept space,
$x_i \in \mathbb{R}^p$ for $i = 1, 2, \ldots, n$ are the bag-of-concepts representations of courses in the training set,
X is an n-by-p matrix where each row is x_i^T
Y is an n-by-n matrix where each cell is the binary indicator of the prerequisite relation between two courses,
i.e., $y_{ij} = 1$ means that course j is a prerequisite of course i, and $y_{ij} = -1$ otherwise,
A is a p-by-p matrix, whose elements are the weights of links among concepts; i.e., A is the matrix of model parameters to be optimized given the training data in X and Y.

The quantity $x_i^T A x_j$ represents the sum of weights of all paths from concepts in course i to concepts in course j through the concept graph A. If there are many concepts in course i that serve as prerequisites to concepts in course j, then the quantity $x_i^T A x_j$ is large and the prerequisite link y_{ij} is set to 1. In this case, the objective is to minimize the difference between 1 (the ideal prediction for a prerequisite relation) and the multiplicative quantity $y_{ij}(x_i^T A x_j)$.
To address the difficulty of classification in this extreme class skewness setting, the problem can be approached as a ranking problem where the objective is to score course and prerequisite pairs (positive examples) higher than unrelated pairs; i.e., the objective would be in this case to maximize the difference of system generated scores given to positive and negative pairs respectively as given by the following minimization problem:

$$\min_{A} \sum_{(i,j,k) \in T} \left(1 - (x_i^T A x_j - x_i^T A x_k) \right)_+^2 + \lambda \|A\|_F^2 \tag{2}$$

where (i, j, k) is a course triplet where course i is a prerequisite for course j and course i and course k are not in a prerequisite relation. In other words, (i, j) is a positive example and (i, k) is a negative example.
One limitation of the above approach is that it also captures overlap and association; i.e., symmetric relations; not only the prerequisite relation which is a directed asymmetric relation. We address this in Section 5.2.
We note that most work involving learning concept relations and concept graph representations for various tasks focus on *asymmetric* measures of similarity whereas our task requires capturing an asymmetric relation: dependency between concepts and documents.
The main contributions of this work can be summarized as follows:

- **introducing** establishing a new research direction in educational data mining at the educational program/curriculum level
- focusing on the **content dimension** by computational formulation of content related problems
- enriching existing **resources** for educational data mining; e.g., educational concept graph in terms of the size (number of concepts), types of relations, and quality/accuracy of links using MOOC and Higher Education Institution (HEI) course data as well as relevant ontologies; e.g., ConceptNet [24] and MetaAcademy [26].

- developing tools for automatically analyzing and **evaluating** an input module/program defined as a set of paths from entry to completion in terms of: learning outcomes coverage, vertical and horizontal coherence, overlap, and difficulty.
- building computational infrastructure for **generating** learning paths and curricula given:
  1. a set of educational objectives and learning outcomes
  2. constraints on amount of overlap, length of the path, and prerequisite structure
- sharing a number of educational datasets and resources, collected from MOOCs and HEI public-access websites, with various annotations including level of difficulty, subject area or specialization, and conceptual dependencies.

4 Content: Three Levels of Abstraction

4.1 Problems in Educational Content Modeling

The following summarizes the main problems of interest in this work. We aim to mine educational data for the following purposes:

1. **Relation Extraction and Link Analysis**: Answering important questions about educational content; e.g., What are the foundation concepts in STEM education? What are some of the more advanced concepts? What are the dependency relations among concepts and topics? How coherent is a given course or module within a course? Which course(s) j are equivalent to a given course i? Given an educational program, does it satisfy a set of learning outcomes? How much overlap and redundancy is there among courses in a given program? Given a learning path, in a proposed educational program, does it satisfy pre-requisite requirements? Does it satisfy intended learning outcomes? Does it cover all educational objectives?
2. **Evaluation**: Analyzing the content in terms of existing patterns, overall quality, and coverage of intended outcomes. Here, the subject of evaluation is the educational content[1].
3. **Content Planning and Generation**: Generating educational content that satisfies desirable properties and specified constraints.
4. **Personalization**: Which courses to recommend to a specific learner given their background and educational attainment? What would be an ideal learning path for a learner who wants to optimize a specific objective; e.g., learning time, coverage of concepts, number of courses, level of mastery?

5 Concepts

Concepts are the representational unit of educational content. Courses are defined in terms of key concepts covered. Modules within courses are defined in terms of closely related concepts. Educational programs are defined and evaluated according to their coverage of specific concepts. Each of the following subsections describes a specific task related to concepts.

[1]Here, the subject of evaluation is the educational content and not the research we conduct to solve a given problem. We discuss evaluation of our methods and empirical results where relevant.

5.1 The Basic Representational Unit

A concept graph is a graph where nodes are concepts and edges are relations between concepts, for example *dependency* relations. The creation of educational concept repositories such as concept graphs using existing resources requires automatic identification of such concepts in unstructured text. Educational data is rich in such elements and the majority of tokens found in course descriptions constitute meaningful concepts. Different choices exist for representing and coding concepts. Some natural choices include words, word stems, lemmas, and (noun) phrases. While stemmed words and lemmas reduce the problem of redundancy and contribute to a more canonical representation, they result in two problems: lack of interpretability of results and reduction of educationally distinguishable terms into a single representation. Using "base" noun phrases to code concepts seems to be a balanced option. It requires, however, parsing the text, and parsing errors may result in noisy output and reduced recall. The choice of base NPs is motivated by noticing that in technical STEM domains, many concepts are described using multiple words (e.g., operations research, differential equations, telescoping sum, big $\mathcal{O}$, etc.). In a corpus containing course descriptions of Coursera courses and more 2322 MIT courses, more than 60% of concepts were bigrams and higher-order ngrams. Additionally, in technical language slight variations in surface form often correspond to different concepts (e.g., operand/operator/operation, difference/differential) which may be lost with stemming. Also, a simple unigram representation of concepts includes many words from grammatical categories that do not typically represent concepts such as adjectives, adverbs, modals, and verbs. Using the Stanford Part-of-Speech Tagger [44] to tag the words in the concept space of [46], we found that more than 40,000 nodes in their concept graph belong to these categories.

In our experiments, using base noun phrases to code concepts improved accuracy in two different tasks: predicting pre-requisite relations among courses (on Coursera, MIT, and CMU data) and predicting dependencies among topics in computer-based assessment. We also report results from expanding the concept space to include higher ngrams.

Our experiments also include distributed representations and word-embeddings of different dimensions trained on large educational corpora. We report results in the following section.

5.2 Predicting Prerequisite Relations

Once concepts are identified, the set of nodes V in the concept graph is formed. The next step is to learn edges E between them. We build on existing work that employs a convex optimization formulation for adjacency matrix completion [46]. In [46], different concept representations were attempted including a word-based concept space and a distributed representation obtained from neural models [4] but was reported to be inferior to other concept space representations. Since the usefulness of distributed representations in different tasks can be sensitive to the choice of pretrained word embeddings, whether they are trainable, and the initialization of unknown word vectors, as reported in [11], we run more experiments to evaluate different choices for concept representation.

First, we expand the original concept space to include higher n-grams found in a corpus of over 7000 course documents. This increases the size of the concept space, originally containing 1,762,583 concepts, by additional 49,514 concepts. Tables 1 and 2 summarize results. We note that the metric gains are small and have not been tested for statistical significance. However, since we see improvement across datasets for more than one metric, this may indicate additional learning from the extra features. Since there is yet a large gap to fill, we follow two different approaches to make improvement: 1. develop an entirely new model and representation and 2. improve the CGL formulation to enable

	Metric		
concept space	MAP	AUC	ndcg@[1:3]
CGL	0.394+-0.013	0.768+-0.021	0.246 0.305 0.363
+freq. ngrams	**0.428+-0.051**	**0.942+-0.006**	**0.351 0.393 0.419**

Table 1: Different concept spaces for CGL on MIT dataset

	Metric		
concept space	MAP	AUC	ndcg@[1:3]
CGL	0.466+-0.037	0.837+-0.020	0.323 0.372 0.426
+ngrams	0.434+-0.025	0.832+-0.022	0.281 0.362 0.400
+freq. ngrams	**0.446+-0.020**	**0.838+-0.014**	0.298 0.356 0.410

Table 2: Different concept spaces for CGL on CMU dataset

better learning. We discuss these approaches next.

5.3 Neural Prerequisite Predictors

We employ deep learning to solve the prerequisite prediction problem. Our model is a deep convolutional neural network with max-pooling and multiple filters of *different sizes* implemented using Keras 1.0 Merge layer. The model has 2 convolutional layers followed by max-pooling followed by two fully connected layers. ReLU activations were used and a softmax for the final layer output.

Different variants of the CNN were used: one with fixed-size convolution windows, and others with multiple convolution windows of size 3, 4, 5, and 7. 128 filters (for each size) were used in each variant. The model was trained for 20 epochs on categorical crossentropy loss using 80% of the data with mini-batch size of 64 and tested on the remaining 20%. Batch normalization was not performed. The variant with two different window sizes (3 and 5) significantly outperformed all other models in all experiments. The model where the embeddings were fine-tuned performed better despite the small size of the dataset.

In transfer learning experiments, the model is trained on data from a specific source and tested on data from another; for example train on MIT courses and test on CMU courses. The model is run for 30 epochs in the transfer learning experiments. Note that CMU dataset contains CS and mathematics courses only whereas MIT dataset includes courses from other STEM fields and Princeton dataset contains mathematics courses only. We have not explicitly trained the model to do transfer learning but to compare with the non-neural baseline which explicitly enables transfer learning by using a large coverage concept graph as an inter-lingua, we include results on transfer learning experiments. We expect some degree of transfer through the use of distributed representations of words in the course description.

We report categorical classification accuracy results. For this specific dataset and task (with tunable embeddings), RMSProp optimizer resulted in better accuracies (86%) compared to Adam (83%). The evaluation metric used is Keras built-in categorical accuracy which is defined as:

```
K.mean(K.equal(K.argmax(y_true, axis=-1), K.argmax(y_pred, axis=-1)))
```

whereas Keras binary classification accuracy uses 0.5 as the threshold to distinguish be-

tween classes and is defined as `K.mean(K.equal(y_true, K.round(y_pred)))`.
For word embeddings, the pretrained GloVe embeddings [34] with 100 dimensions were used and tuned. Vectors of unknown words were randomly initialized.
The dataset consists of course descriptions (often including a detailed syllabus) from three different universities: MIT, CMU, and Princeton. This is a subset of the data used in [46]. The dataset also includes prerequisite links between courses offered by the same institution. To frame the problem as a classification problem, textual descriptions of pairs of courses are concatenated. When the first course is a prerequisite for the second, we get a positive instance and we get a negative instance otherwise. The classification problem can be viewed as judging whether a document is coherent or not where a coherent document presents prerequisite content earlier.
For comparability with the baseline, which did not employ syntactic features, a natural choice is a generic neural model. The model we implemented is based on the state-of-the-art CNN for sentence classification [21] with several modifications in filter design and choice of hyperparameters. For comparability with the baseline, the model is evaluated on AUC. We used the same dataset of course textual descriptions in [46] but curated it for the neural classification task. We note the extreme skewness of the data since only a very small subset of course pairs are positive instances of the prerequisite relation. After creating pairs from the set of 2461 courses, we randomly undersample the negative class such that the resulting distribution is still comparable to the baseline. This results in a dataset where the positive class constitutes 26% of the data. We finally shuffle the instances to avoid bias in learning. We outperform the baseline by a statistically significant margin in all datasets. Although the baseline is not neural, we note that it is based on a large-scale optimization and has nearly a billion parameters in the original space whereas our neural model has 2.8M parameters. Table 3 presents accuracy results of the neural models.

model	categorical accuracy
CNN fixed filter size fixed embeddings	0.4905
CNN variable filter size fixed embeddings	0.8141
CNN fixed filter size trainable embeddings	0.5112
CNN (4) variable filter size trainable embeddings	0.8417
CNN (2) variable filter size trainable embeddings	0.8571

Table 3: Performance of different neural models on prerequisite prediction

Results The best performing model is evaluated using AUC on different datasets. Significant improvement over the CGL baseline is achieved. Table 4 shows AUC results.

dataset	CGL	CNN	Δ
MIT	96%	94%	-2
CMU	79%	97%	+18
Princeton	92%	97%	+5

Table 4: CGL vs. neural model: AUC

Table 5 presents transfer learning results where the best performing model is trained on data from a given source and tested on data from a different one.

source	target	CGL	CNN
MIT	Princeton	72%	66%
MIT	CMU	70%	82%
CMU	Princeton	NA	69%
Princeton	CMU	NA	63%

Table 5: Transfer Learning Results: AUC

5.4 From Concept Graphs to Concept DAGs: Association vs. Dependency

Current approaches to concept graph learning do not yield a DAG structure [46, 15]. A concept graph encodes dependency relations between concepts and hence should be free of cycles. Ideally, a concept graph is a directed, acyclic[2], transitive graph; i.e., a *tournament*. When such a graph is learned from data that is limited in amount and noisy in nature, it is not expected to be a perfect tournament. However, a directed acyclic graph can be easily transformed into a transitive one, but producing a directed acyclic structure remains a challenge.
We propose to modify the formulation of the concept graph learning objective to include a term that forces random walks of the graph that begin and end at the same node to be small or zero. The original formulation for the classification approach is [46]:

$$\min_A \sum_{i,j} \left(1 - y_{ij}(x_i^T A x_j)\right)_+^2 + \lambda \|A\|_F^2 \tag{3}$$

where n is the number of courses in a training set,
p is the dimension of the concept graph,
$x_i \in \mathbb{R}^p$ for $i = 1, 2, \ldots, n$ are the bag-of-concepts representation of a course in the training set,
X is an n-by-p matrix where each row is x_i^T
Y is an n-by-n matrix where each cell is the binary indicator of the prerequisite relation between two courses,
i.e., $y_{ij} = 1$ means that course j is a prerequisite of course i, and $y_{ij} = -1$ otherwise,
A is a p-by-p matrix, whose elements are the weights of links among concepts; i.e., A is the matrix of model parameters to be optimized given the training data in X and Y.

The proposed formulation is:

$$\min_A \sum_{i,j} \left(1 - y_{ij}(x_i^T A x_j)\right)_+^2 + \lambda \underbrace{\sum_{k=2}^{n} tr(A^k)}_{\text{new term}} \tag{4}$$

Each of the $n-1$ terms in the new sum corresponds to cycles of length k where k ranges from 2 to n. The first term in the sum subsumes the matrix Frobenius norm which is the regularization term in the previous formulation. For even k, the corresponding term is known to be convex. Convexity of odd terms is not known. The optimization can be approached by considering even terms only or using a combination of convex and non-convex optimization methods for the odd terms. To force a sparser DAG, the following

[2]Acyclicity implies irreflexivity and asymmetry.

formulation is proposed:

$$\min_{A} \sum_{i,j} \left(1 - y_{ij}(x_i^T A x_j)\right)_+^2 + \lambda \underbrace{\left|\sum_{k=2}^{n} tr(A^k)\right|_1}_{\text{new term}} \tag{5}$$

A similar modification is proposed for the ranking approach as well.

5.4.1 Convexifying the DAG Learning Objective

The proposed formulation can be expressed as a sum of a convex function (loss + convex regularizers) and a non-convex but differentiable function (sum of regularization terms with odd k). Note that the trace of a square matrix is a univariate function of the eigenvalues of A.
By definition, the characteristic polynomial of a square $p \times p$ matrix A is given by:

$$p(t) = det(A - t\,I) = (-1)^p(t^p - tr(A)t^{p-1} + \ldots + (-1)^p\, det(A)) \tag{6}$$

but

$$p(t) = (-1)^p(t - \lambda_1)\ldots(t - \lambda_p) \tag{7}$$

, where the λ_i's are the eigenvalues of A. Therefore,

$$tr(A) = \lambda_1 + \ldots + \lambda_p = \sum_{i=1}^{p} \lambda_i \tag{8}$$

Without loss of generality we can assume that p is even. We can write the odd terms in the new formulation as sums of squares of the eigenvalues of A. By introducing new variables, λ, we have convexified the DAG learning objective. The new objective is convex in A and λ's and the two parts are connected by the constraint $tr(A) = \sum_{i=1}^{p} \lambda_i$. This is still an intractable problem due to the large number of parameters. For a vocabulary of 100K terms, the loss term alone has a billion parameters. Work to formulate the problem in the dual space is in progress.
Significance: this will not only solve the problem of link prediction and cycle elimination in the concept graph learning task but may be a first step towards adjacency matrix completion for Directed Acyclic Graphs. To our knowledge, no formulation has been proposed or studied for matrix completion of DAG structures.

Although being directed and having cycles are two distinct properties of graphs, they are related in the following sense. If the formulation of the learning objective fails to capture directionality, we expect to find more cycles in the resultant graph. The following approaches are proposed to capture directionality at the course level:

- Use set intersection and difference to predict the correct direction of a pre-requisite relation. If two courses c_1 and c_2 are expected to be in a prerequisite relation and $|c_2 - c_1| \gg |c_1 - c_2|$ then we expect c_1 to be a pre-requisite for c_2. This is based on the intuition that a course uses concepts from a prerequisite course but also introduces new concepts. Note that we use this as a sufficient (not a necessary) condition for directing the prerequisite relation. Similarly, if many concepts in c_1 appear in c_2 but not vice versa, we predict that c_1 is a prerequisite for c_2. We propose to empirically evaluate these assumptions on course pairs from real datasets.

- Compute difficulty scores of two courses expected to be in a prerequisite relation. Use the difficulty comparison to find the direction of the prerequisite relation. For this task, a regressor is trained to predict the difficulty score of a course. Course codes are used as a proxy to difficulty labels (higher codes suggesting more difficulty), and features include lexical, syntactic, and text complexity features. Several datasets are merged for this task including the course catalogue from different HEIs as well as Coursera courses. For Coursera courses, the ordinal "intended audience" information is used as a difficulty label.

5.5 Enriching Existing Concept Graphs

Previous work on educational concept graph learning was limited to identifying one type of relation among concepts: dependency. To analyze and generate curricula, it is important to identify other types of relations among concepts. These include: equivalence[3], instantiation (example-of), and subsumption (includes or consists-of). Both content and structure of educational data can be leveraged to learn these relations among concept. We extracted over 20,000 relations of these (additional) types from Courera data using structure and construction matching. More relations can be learned from observed equivalence and overlap relations among courses in different datasets.

5.5.1 Enlarging the Concept Space

Detailed course data was collected to enlarge the size of the concept space. This includes concepts from detailed course syllabi collected from Coursera STEM courses. This dataset includes, in addition to course descriptions, detailed syllabi of courses organized by topic. This significantly increases the number of nodes in the concept graph. However, these nodes are not all created equal. Although every node denotes a concept, these concepts differ in their difficulty, granularity, and position in a hierarchy of topics.

5.5.2 Including Additional Relation Types

Identifying different types of relations among concepts; i.e., annotating the edges with different types of labels in addition to numeric scores, amounts to labeling nodes with relative levels of granularity and complexity. If the *geometry* and *calculus* nodes are in a part-of relation with the *mathematics* node then the latter is a coarser concept. Eventually, a hierarchy of concepts can be extracted from the flat graph structure. This hierarchy is important when planning a curriculum as it distinguishes broad topics from individual concepts.

5.5.3 Tagging Concepts with Difficulty Level

An important aspect of personalized curriculum planning is customized depth and level of difficulty in a way that matches the learner's background and intended level of mastery. The overall level of difficulty of a learning path is a function of the level of difficulty of units within this path which in turn depends on the level of difficulty of the concepts in each unit and how much overlap is there between the different units (repetitiveness). We use different information in different datasets as a proxy to the level of difficulty. For HEI data, we use the course code and level (graduate or undergraduate) while for Coursera data we use the "intended audience" field to tag concepts with a level of difficulty. This

[3]Compare to synonymy among words. Equivalence of concepts includes synonymy but also extends it to apply to more complex notions of equivalence; e.g., equivalence of EM and ICE algorithms for the exponential family.

provides partial labeling of documents (and therefore concepts) in the different datasets, we propose to complete the tagging of other concepts in the concept graph using KNN classification where the distance is measured in number of edges. In the course space, course concept overlap can be used as a measure of distance. Alternatively, we propose to train a regressor that predicts the relative level of difficulty for a given pair of courses given concepts as features and previously annotated pairs as input data.
In addition to link features in the concept space, we propose to investigate the usefulness of node specific lexical, syntactic, and semantic features for predicting its level of difficulty. Such features include the length of the concept in words, (average) length of words constituting the concept, number of word senses, and frequency of the concept in addition to structural features given by the graph-degree metrics of the concept page on Wikipedia.

In order to improve the concept graph coverage and links, we created a rule-based parser to extract "chains" of concepts from Coursera course syllabi. From a collection of 1,007 courses in STEM domains, 766 "gold" paths were extracted with an average length of 7 concepts. This results in additional 20,000 relations that concept graph learning can be initialized with. Figures 4 and 5 show examples of these paths.

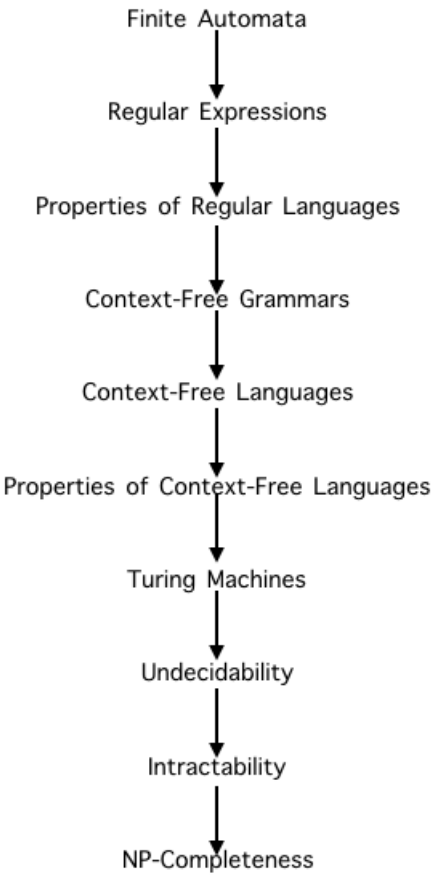

Figure 4: A concept path in automata theory extracted from Coursera data

Figure 5: A sample of concept chains extracted from Coursera data

Since manual creation of concept prerequisite relations is an expensive task, we explore other sources of concept-level dependencies including explicit dependency relations found in textbook prefaces, such as those in Figure 6 and the new MetaAcademy project [26].

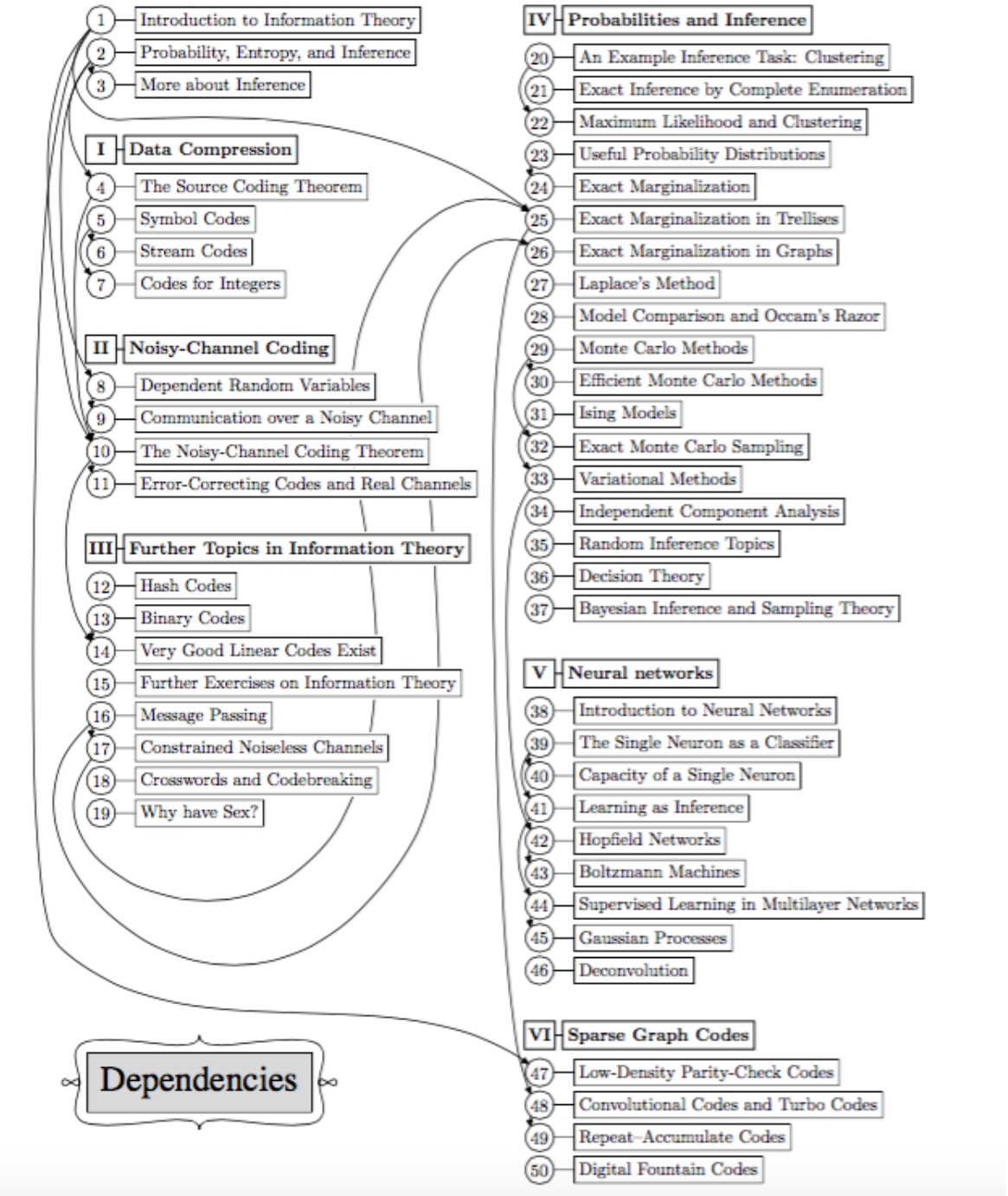

Figure 6: Example of sources for gold concept-level dependencies. Source: *Information Theory, Inference, and Learning Algorithms* by David J.C MacKay. Cambridge University Press

5.6 Leveraging Expert Knowledge in Learning of Concept Graphs

So far, we have described a purely data-driven approach for learning a universal concept graph. We propose to also leverage the knowledge in existing resources and ontologies and combine that knowledge with our approach to produce a concept graph that is robust to noise in the data (1) and consistent with existing knowledge about topics and their relations in the educational domain. In this approach proposed in [14], the pre-trained concept vector is post-processed to be close to vectors of concepts that are related to it in a given ontology. For example, in ConceptNet [24], *algebra* is linked to *pure mathematics*, *algebraic geometry*, and *matrix algebra*. The goal is to have a concept vector for*algebra* that is close in distance to vectors of related concepts while optimizing vector entries learned from data. This is achieved by retro-fitting (and possibly counter-fitting) distributional concept profiles to existing knowledge about concept relations.
Let $V = \{c_1, \ldots, c_n\}$ be the set of concepts in the concept graph G and $\hat{E}$ be the relations between the concepts as learned from the data. Let $G_{ontolgy}$ be the graph representing a given ontology of concepts with nodes V and edges E. Let matrix A consist of vectors representing concepts in G: $\hat{v}_{c_1}, \ldots, \hat{v}_{c_n}$. For a concept c_i, we want to learn a vector representation v_{c_i}, that is close to related concepts as learned from the data $\hat{v}_{c_i}$ and is close to vectors of related concepts in the ontology v_{c_j} such that $(i, j) \in E$. This amounts to minimizing the following objective function [14]:

$$f(Q) = \sum_{i=1}^{n} \left[\alpha_i \left\| v_{c_i} - \hat{v}_{ci} \right\| + \sum_{(i,j) \in E} \beta_i \left\| v_{c_i} - v_{c_j} \right\| \right]$$

where α and β control the relative strength of each component and are externally tuned.

We propose to implement this approach for directed and undirected concept graphs and test each in different tasks in the concept and course spaces including: educational concept similarity, educational concept difficulty prediction, course pre-requisite prediction and course topic categorization task (which broad topic/field(s) does the course belong to).

5.7 Concept Graph Summarization and Link Analysis

While the educational concept graph is a useful resource for several downstream tasks, it could also be used to answer key questions about corresponding domains. For example, what are key skills required in STEM education? What are the common and foundation concepts required across STEM specializations? What STEM specializations are most closely related? What new connections are observed between different topics/concepts? What concepts are usually taught together? What are the most frequent concepts in a specific field? What are the most commonly used methods and theoretical tools?
To answer these and similar questions, we propose a comprehensive link analysis on the universal concept graph. This includes degree analysis, running pageRank on the concept space, running various clustering algorithms to find major connected components of the graph and cluster overlap patterns, and using graph summarization methods to find key structures in the graph [22]. We also propose to use graph traversals methods to find interesting subgraphs such as cliques (in the undirected case), near-cliques, stars, chains, and bipartite cores (if any). We propose to compare findings with perceptions of educators and learners and the general practice of curriculum design in corresponding fields. We have conducted preliminary link and centrality analysis on concept graphs constructed from different datasets. We discuss various aspects of this study next.

5.7.1 Data

Data for this task was obtained from Coursera website through its API [10]. Coursera is one of the largest providers of free MOOCs offering more than 1000 courses with a total number of enrollments exceeding 12 millions (compare, for example, to 400 courses offered by edX and 71 by Udacity) [9, 45, 13]. More than 84% of courses offered by Coursera has English as the language of instruction. At least 50% of courses offered by Coursera are STEM courses as shown in Figure 7. For link analysis, we considered a subset of Coursera that lies in the intersection of STEM courses and courses taught in English. The exact number of courses included in the analysis is 766.
In addition to function words, a list of common words and phrases was compiled based on a statistical analysis of the corpus using the CMU-Cambridge Statistical Language Modeling Toolkit v2 [40]. The data was further refined by removing these. The complete list of common words and phrases is given in the appendices. These words and phrases can be used as features for identifying prerequisite-related content in educational text in general. Finally, the Stanford parser [25] was used to extract base NPs. On the other hand, the universal set of concepts in [46] was obtained by using words extracted from course descriptions to query Wikipedia and use the top retrieved documents to extract relevant categories and then use content words and wikipedia categories to represent the concept space. Their approach is based on a unigram representation of concepts. Furthermore, the words representing concepts are stemmed as a standard preprocessing step along with removal of common words. Two main limitations of their approach is loss of multi-word concepts and collapsing semantically different concepts into one code. In technical language slight variations in surface form often correspond to different concepts (e.g., operand/operator/operation, difference/differential, derivative/derivation).

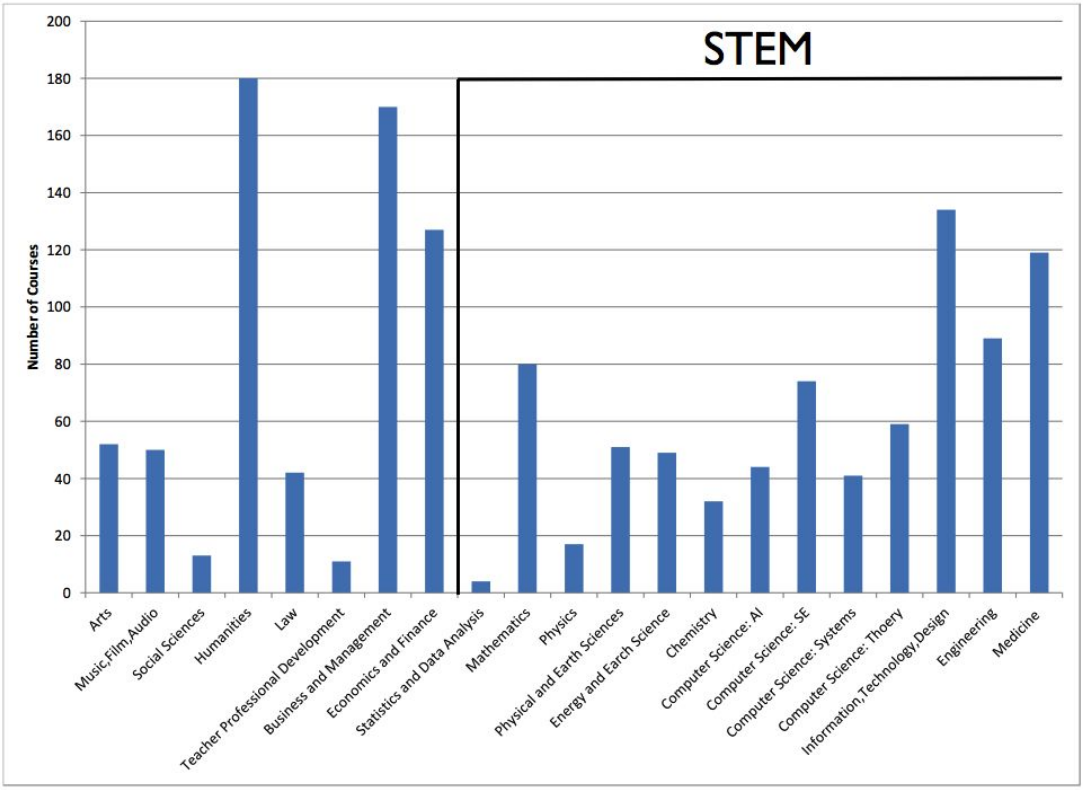

Figure 7: Distribution of Coursera Courses by Specialization

The following table summarizes the different networks constructed for the graph analysis task and their corresponding datasets. All networks are weighted and all are single-mode networks.

Network	Coursera	MIT	CMU	LTI[4]
Type	Directed	Directed Singed	Directed Singed	Symmetric
Nodes	20K	1.7M	1.7M	25
Edges	52K	$\mathcal{O}(10K)$	$\mathcal{O}(K)$	509
Data Source	Coursera STEM Courses	MIT STEM Course Catalgoue	CMU CS & Math Course Catalgoue	LTI Course Catalgoue
Construction Method	1-to-all bipartite edge construciton	Adjacency Matrix Completion Learning-to-Rank Optimization		Overlap Computation

5.7.2 Concept Networks: Global Metrics

More than 95% of nodes in CMU and MIT concept graphs are **isolates**; i.e., with indegree and outdegree = 0. Since many stemmed words and terms retrieved from queries to Wikipedia do not represent concepts, they are are not expected to participate in concept dependency relations. MIT network, the largest of the three, has only one large connected component. Table 6 presents a node and degree summary for different versions of

	-ve binary	-ve weighted	+ve binary	+ve weighted
nodeset	1761276	1761276	1761276	1761276
Isolates	1746369	1746369	1747184	1747184
deg dev	212.694	0.27017	97.5857	0.279443
interval	[0,603.75)	[0,2.60946)	[0,452)	[0,2.3664)
freq.	1755074	1760043	1757111	1759895

Table 6: Node and Degree Summary for MIT Concept Networks

the MIT concept graph: positive, negative, weighted and binary. This table shows that the network is extremely fragmented and has very low density < 0.0005. The extreme fragmentation of the graphs suggests that edge-based graph traversal method would fail when searching or navigating this graph. To address this problem, we propose to seed concept graph learning with links in extracted concept paths.

The concept graph learning algorithm in [46] claims to learn *directed dependency relations* between concepts. If concept i is a prerequisite for concept j, then ideally edges in both directions should exist with weights 1 and -1, for prerequisite and reversed prerequisite relation, respectively. In network terms, the adjacency matrix of the network should be as close as possible to the negative of its transpose: $A = -A^T$, however when computing the correlation between the concept graph and its transpose it is found to be -0.0052 when it should ideally be equal to (or close to) -1. Similarly, the correlation between the course prediction matrix Y and its transpose should be (close to) -1. When actually computing this quantity, the correlation is found to be 0.0272. When reversing the order of course pairs in the test set in CGL, the algorithms makes similar predictions and performance degrades which indicates its inability to distinguish the direction of the relation.

This suggests that the algorithm is learning some kind of symmetric association between content units and picking up prerequisite relations in the process since overlap and association is actually found between courses and their prerequisites.

These findings emphasize the importance of capturing the directionality aspect when learning links and minimizing cycles in the learned concept network.

The graph obtained from CMU training data contains 1955 nodes with **self-loops**. The graph also contains cycles of various lengths. Similarly, the graph based on MIT data contains 7837 nodes with **self-loops** and has cycles of various lengths. When running a cycle detection program on these graphs, results are positive for every cycle length tested, *except 2*. This is because the regularization term $\|A\|_F^2$ forces random walks of length 2 that begin and end at the same node to have weight zero and hence to be eliminated.

When links represent dependency relations between concepts both self-loops and cycles represent irrational cyclic logic. This also breaks the inference algorithm which attempts to use this graph to predict prerequisite relations between new pairs of courses. The prevalence of cycles in this network explains the suboptimal performance measured as Mean Average Precision on the task of predicting prerequisite relations for new pairs of courses as reported in [46].

5.7.3 Concept Networks: Node Centrality and Edge Metrics

The following tables list the highest value dependency relations for the concept networks. yclic dependencies are underlined. Authority and hub nodes obtained from running pageRank on the three graphs are also given and compared.

concept	dependent concept	relation-strength
complex	seri	0.2594
integr	function	0.2491
data	statist	0.2296
function	seri	0.2212
learn	recit	0.2208
probabl	integr	0.2184
random	surfac	0.2084
theorem	integr	0.2083
graph	matrix	0.2027
surfac	seri	0.1991
theori	integr	0.1975
integr	deriv	0.1934

Table 7: High-score links: CMU Concept Graph

concept	dependent concept	relation-strength
algebra	group	0.5342
manag	manag	0.5276
cell	recombin	0.4804
patent	patent	0.4568
teach	tutor	0.4223
biolog	recombin	0.4146
cell	biochemistri	0.4140
chemic	chemic	0.4107
anchorencod	seri	0.3656
theri	field	0.3653
algorithm	matric	0.3616
physic	field	0.3575
biolog	rna	0.3561
biolog	cell	0.3535

Table 8: High-score links: MIT concept graph

concept	dependent concept	relation-strength
quantitative_research_methods	data	0.0014
mathematics	algorithms	0.0014
computer_science	security	0.0012
statistics	operations	0.0012
quantitative_research_methods	statistics	0.0011
programming_experience	programs	0.0011
nature	plants	0.0011
mathematics	data	0.0011
MOOC	organization	0.0011
mathematical_thinking	auctions	0.0009
computer_science	hardware	0.0009
undergraduate_introductory_cell_biology	systems	0.0009
MOOC	content_strategy	0.0009
AP_AB	product	0.0009
mathematical_thinking	agents	0.0007
calculus	functions	0.0007
computer_science	model	0.0007
quantitative_research_methods	inferential_statistics	0.0007
molecular_biology	molecular_evolution	0.0007

Table 9: High-score links: Coursera concept graph

We observe cyclic dependencies even among the high-value relations extracted from MIT concept graph. The quality of relations extracted from MIT data seems to be better compared to relations extracted from CMU data. This may be explained by the size and topic coverage of the two datasets. We also note that relations extracted from CMU and MIT data reflect an *association* between concepts but not necessarily a *dependency* relation. A large number of these relations was obtained from a simple bigram frequency analysis using the CMU-Cambridge Statistical Language Modeling Toolkit [40]. On the other hand, relations extracted from Coursera data seem to represent stronger dependencies. We also notice that concepts in MIT and CMU graphs are more specific while concepts from Coursera data, at least in the high-value relations, are broader topics. Another distinction between the two types of concept graphs, Coursera and HEI, is the larger overlap between the set of concepts and the set of prerequisite concepts in the latter case. This is expected given the different representation of prerequisite material in the two cases: a textual description of recommended/required background in Coursera and another course description in HEI data. This can also be explained by the skewed distribution of degree in the MIT and CMU extremely sparse networks while a more uniform edge weight distribution is observed in the denser Coursera network

Results of running pageRank on the concept graphs are summarized in Tables 10 and 11 in terms of the top authority and hub nodes. Authorities and hubs across these different networks show that mathematical knowledge and skills such as modeling, algorithmic thinking and quantitative data analysis are key concepts in STEM domains. As in the previous link analysis, we observe that node hubs and authorities are given in broader terms in Coursera case and in more specific terms in MIT and CMU networks. This may agree with the fact that Coursera courses are intended for a wider audience and therefore present concepts and requirements, at least in course descriptions, in more general terms.

CMU	**MIT**	**Coursera**
model	system	mathematics
theorem	engin	statistics
kinemat	theori	algebra
comput	model	quantitative_research_methods
program	equat	programming
analysi	cell	MOOC
data	energi	molecular_biology
graph	mechan	mathematical_thinking
linear	biolog	computer_science
logic	design	high_school
learn	comput	interdisciplinary_perspective
random	tissu	clinician
system	analysi	java
integr	introduct	AP_AB
robot	method	communication_networks
percept	algebra	R_programming
dynam	quantum	high_school_level_science
theori	wave	bioelectricity
differenti	project	undergraduate_introductory _cell_biology
proof	structur	frehman_chemistry
introduct	process	computing_proficiency
applic	linear	dental_background
recit	cont	calculus
ii	law	literacy
algorithm	present	data_structures
probabl	function	(basic)engineering_mechanics
distribut	algorithm	Neuroscience

Table 10: Centrality and Concept Importance: top 30 authorities

CMU	MIT	Coursera
linear	exam	data
recit	cover	applications
seri	integr	science
advanc	linear	information
search	forc	problem
equat	variabl	algorithms
integr	comput	process
prog	order	practice
matrix	theorem	methods
recurs	matric	field
dimens	angular	design
complex	field	time
memori	potenti	challenges
test	cont	health
represent	review	terms
machin	fourier	programs
tree	complex	models
comput	momentum	principles
random	seri	organization
binari	function	material
part	electr	management
probabl	equat	motion
proof	recit	theory
basic	exponenti	behavior
method	rule	MOOC
function	time	opportunity
theori	infer	life
orthogon	system	analysis
direct	biochemistri	English
solv	respons	technology

Table 11: Centrality and Concept Importance: top 30 hub nodes

5.7.4 Groups in the Concept Graph

Coursera network was clustered, on network structural features, using k-means which produced 6 groups. The local clustering coefficient, averaged over nodes, is 0.014 while the global clustering coefficient is 0.032. This network is denser and more connected compared to MIT and CMU networks created using adjacency matrix completion. Figure 8 shows the block model of the groups. If we group Coursera Computer Science subfields into one group, group Mathematics and Statistics and Data Science into one group, and group Physical and Earth Science, Physics and Energy and Earth Science into one group we get exactly 6 groups. We hypothesize that the largest cluster in k-means corresponds to the largest Coursera specialization which is the computer science course collection. To examine the correspondence between the clusters produced by k-means and the actual STEM areas, we would need to sample from each cluster for cluster identification. The network block diagram is a low-dimensional representation of the network and summarizes

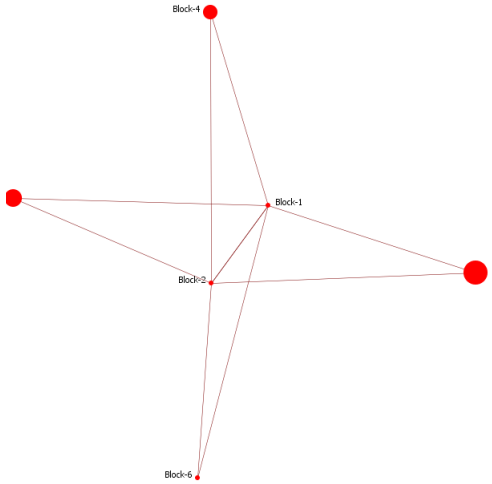

Figure 8: Block model of clusters in the Coursera concept graph

the interaction between groups in the network. This is useful for understanding topic and module-level relations in educational data.

5.8 Evaluation

Since the educational concept graph is used for various prediction tasks at the course and curriculum levels, we primarily propose extrinsic evaluation methods to assess the quality of the concept graph and the links therein. We describe those tasks in more detail in the following sections. Additionally, we propose human evaluations of random samples of the learned relations as well as benchmarking against existing relation repositories [24, 26].

6 Courses

6.1 Representation

Courses are educational units covering specific subjects for the duration of an academic term. In self-directed or personalized learning, however, the length (or duration) of a course can vary and the definition of the course depends primarily on the coherence of its content. The coherence of the content depends on the relatedness of concepts within the course (existence of edges between nodes) and the strength of those relations (edge weight). In the following section, we explain how coherence is formalized and optimized. Throughout this section we may use terms course, unit, and module interchangeably

6.1.1 Planning and Generation: Coherence

Building on the work of [42] which focuses on the news domain, we formalize the characteristics of a well-designed course in terms of its *coherence*. The main idea is to evaluate the coherence of multiple paths connecting two concepts and then select the most coherent path. The approach could be used for both **evaluating** and **generating** coherent content. To our knowledge, course coherence has not been computationally formalized and no method has been proposed yet to automatically score course coherence. We aim

to uncover hidden connections between two elements in the content space to form a logical coherent module. Given two topics or modules, can we automatically find a coherent chain of concepts linking the two together? This question could be asked at the course or curriculum levell. More specifically, we could ask about the coherence of a course (chain of concepts) or the coherence of a curriculum.
Coherence of content has three related yet technically distinct meanings:

- **Gap-free**: a coherent module is well-connected and does not have gaps from one topic to the other.
- **Correctly ordered**: topics in a coherent module are ordered such that prerequisite content appears before content that depends on it. This is referred to as *vertical coherence*.
- **Consistent across alternative paths**: when there are multiple paths between the two modules (or from entry to graduation), the different paths should be consistent in coverage of learning outcomes. This is referred to as *horizontal coherence*.

We focus on the first and second aspects of coherence in this section.

Coherence Analysis We formulate the coherence of a sequence of topics as follows. Let $\mathcal{C}$ be a set of educational units (courses or topics), and $\mathcal{W}$ a set of features (words/concepts). Each unit contains a subset of $\mathcal{W}$. Given a chain $(c_1, \ldots, c_n)$ of topics from $\mathcal{C}$, we can estimate its coherence from its concept activation patterns using two alternative approaches:

$$Coherence(c_1, \ldots, c_n) = \min_{i=1,\ldots,n} \sum_{w} \text{LinkStrength}(c_i, c_{i+1}|w) \tag{10}$$

$$Coherence(c_1, \ldots, c_n) = \max_{activations} \min_{i=1,\ldots,n} \sum_{w} \text{LinkStrength}(c_i, c_{i+1}|w)\mathbf{1}(w \in c_i, c_{i+1}) \tag{11}$$

These formulations capture the following important characteristics of coherence:

- Defined in terms of the **important** key concepts not just the mere overlap or similarity of two consecutive modules in the chain. This is captured by the LinkStrength term. This term is large;i.e., link is strong, under two conditions: if the two modules are strongly related, and w is important for connecting them. The specific definition of Link Strength is given later in this section.
- Depends on the strength of the **weakest link**. This is based on the intuition that "a chain is only as strong as its weakest link." Therefore, instead of summing over all links which can be misleading since the score of the very strong links can dominate the score of the weaker links producing an overall misleading high coherence score. Therefore, a more reasonable objective is to consider the minimal transition score instead of the sum. This is captured by the min operator.
- The second formulation additionally encourages a **global theme** which is realized by longer stretches of topic activation. This formulation penalizes incoherent chains which are chains that suffer from topic gaps and disconnectedness where topics appear and disappear throughout the chain without maintaining a topic focus. This aspect is captured by the max over activations.

This coherence equation could be turned into a linear program where mix, max, and word activation requirements are translated into constraints in the linear program. The computation of the link strength is explained next.

Link Strength Computation For determining the keyword importance graph between modules, a bipartite directed graph is constructed where edges link modules to concepts that appear in them and vice versa. When gold conceptual dependencies are available, we use that information to link courses to requisite concepts.
First, a bipartite directed graph, $G = (V, E)$ is constructed with courses (or modules) as one vertex subset and keywords/concepts as the other vertex subset; i.e., $V = V_{\mathcal{C}} \cup V_W$. For every concept w in c (or known to be important in c), edges in both directions are added. An example is shown in Figure 9.

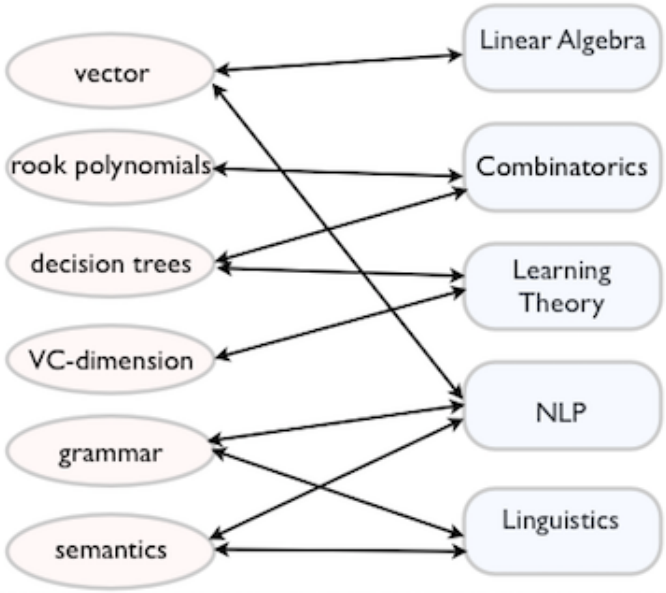

Figure 9: Bipartite construction of course c and keyword w link graph

Edge weights represent the strength of the correlation between a module and a word/concept. The link strength term LinkStrength$(c_i, c_j|w)$ captures the strength of the connection between the two modules c_i and c_j conditioned on w. The value of this term is high if the two modules are strongly related and w plays a role in this relation. In this case, a random walk starting from c_i should reach c_j with high probability. The stationary distribution is the fraction of the time spent at each node.

$$\prod_i(v) = \epsilon.\mathbf{1}(v = c_i) + (1 - \epsilon) \sum_{(u,v)\in E} \prod_i(u)P(v|u) \tag{12}$$

where $P(v|u)$ is the probability of reaching node v from node u. In order understand the importance of w in these walks, w is turned into a sink node; i.e., once it's reached there are no outgoing transitions. We consider a probability distribution $P'(v|u)$ which is the same as $P(v|u)$, except that w is now a sink node. Similar to the definition in Equation 12, $\prod'_i(v)$ is the stationary distribution for the new graph in which w a sink node. The intuition is that if w is a key concept, the stationary distribution of c_j would decrease because without the concept that links the two courses, c_j becomes unreachable from c_i. We expect to see a more significant decrease for key concepts.
The importance of w in linking c_j is defined as the difference between the two distributions: $\prod_i(c_j)$ and $\prod'_i(c_j)$.

Coherent Sequence Generation Using coherence scoring, several approaches could be explored for the generation problem. We briefly list some next:

- Local greedy search: start with a candidate topic sequence and iteratively move to another sequence by changing an edge or node at a time to maximize the coherence score defined earlier. This local search method is known to suffer from local optima but it is one option to investigate if combined with a good starting sequence.
- Joint optimization of score and sequence: translate the following requirements into constraints in a linear program* and then solve:
  - *chain restriction*: start and end the sequence with the desired nodes, limit the number of nodes in the sequence to a target number, and ensure a chain structure: every node (except start and end) has one incoming and one outgoing edge
  - *activation restriction*: restrict the number of active features (keywords or concepts) per course and per chain. If a concept is active in a transition, the transition has to be active; the concept activation level cannot exceed its activation level in the course/unit from which it originated; i.e., the concept's *source course* has its maximum activation.
  - *minmax objective*: find the minimum score edge over all active edges, maximize that (min weight) edge; i.e., strengthen the weakest link in the sequence

6.2 Evaluation

We propose to evaluate the above coherence method by comparing it to a shortest-path baseline. We also plan to collect human expert judgements to evaluate the output sequences in three aspects: relevance, coherence, and redundancy.
We also propose to take advantage of program accreditation information publicly available [1]. Accredited programs (curricula) are evaluated by human experts and are expected to exhibit coherence. Experiments in the **analysis** problem can proceed by scoring accredited programs and system generated sequences and then compute the Kendall Tau distance between the system generated sequence and the accredited program sequence as a gold standard.

In addition to course descriptions and syllabi from different universities, data for concepts and course prerequisite links can be obtained from educational reports of accreditation bodies (ABET, ISO-Education) and professional organizations; i.e., CS educational program guidelines published by ACM and IEEE-CS [2]. These can be used, with proper "featurization" for concept-course relation extraction and therefore the construction of the bipartite graph explained above.

7 Curricula

7.1 Definition and Representation

An educational program or curriculum is a planned sequence of courses or modules designed to (1) cover specific **learning outcomes** defined as a conjunctive set of concepts to be mastered, (2) define a learner's progress towards achieving a broad set of **educational objectives**, and (3) be typically completed within a particular **timeframe**. Typically, a curriculum consists of required (core) and optional (elective) components. At each step of the progress, the learner can make a choice from a set of courses/modules given that prerequisites are fulfilled. There is usually more than one possible path from entry to graduation that cover the intended learning outcomes, satisfy the prerequisites, and

can be completed within the duration of the program. Next, we attempt to answer the question of how to navigate the space of modules while meeting the learning objectives. We also formalize the notion of "optimal" learning path given desirable properties of a curriculum such as coherence, progressive difficulty, and coverage of intended outcomes. A curriculum includes, in addition to "dependencies" between modules, other relations such as similarity, equivalence (→mutual exclusion) and co-requisites. Modeling these additional relations is of special importance given the fragmentation of automatically learned concept dependency graphs.
We formalize characteristics of well-designed modules based on properties identified in the literature of instructional design [30, 35, 17, 3] as follows:

- **Cohesion** A module in a curriculum contains closely related inter-dependent concepts which form cohesive units of study; e.g.; finite automata ⇔ regular languages, force ⇔ joule
- **Isolation**: concepts ⊂ different modules should be as independent as possible
- **Unity**: a concept should be covered in a single unit

We propose applying clustering algorithms using textual and graph-structural features of concepts to create cohesive modules. Since we expect noisy clusters, we propose to de-noise the relations in the obtained clusters by training a classifier to predict whether an edge is placed correctly between two concepts and whether the edge is from concept u to concepts v or from concept v to concepts u. We propose network based features to be used in the edge classification task. These features can be extracted from the concept graph directly or from wikipedia graph of the concepts , where in-degree/out-degree are computed as the article incoming/outgoing links, respectively. Other features like the number of languages the article is available in can be used as predictive features. Fundamental concepts are more likely to have pages in multiple languages. Also the number of Wikipedia categories per concept wiki-page can be used as another useful feature. Basic and fundamental concepts tend to have a smaller number of categories. The following is a list of graph features for the edge de-noining task:

- **In-degree**: higher for fundamental concepts; the intuition is that if the in-degree of $u \gg$ in-degree of v than u is likely to be the fundamental concept; hence the prerequisite concept as many more other concepts refer to it
- **Out-degree**: higher for advanced concepts; the intuition is that if the out-degree of $u \gg$ out-degree of v than u is likely to be the more advanced concept; i.e., u is the dependent concept and v is the prerequisite as u reference many more concepts compared to v
- **Common neighbors**: this features captures the relatedness of concepts u and v. For every edge (u, v), the feature counts the number of common neighbors. If the two nodes u and v are structurally equivalent, we may conclude conceptual synonymy (e.g., shattering coefficient≡growth function)

Once cohesive units of content with minimal overlap and desired coverage are formed, we order them in a sequence according to one or some of the following criteria:

- **Prerequisites**: sequence modules in any order as long as prerequisite constrained are satisfied
- **Difficulty**: modules containing easier concepts are introduced earlier
- **Early Foundation**: group and introduce all foundational concepts in prerequisite courses early in the curriculum
- **Locality of References**: if c_j depends on c_i, $\min(dist(c_j, c_j))$ in the curriculum; i.e., order modules such that the distance between prerequisite and dependent concepts is minimized.

We put the above together and formulate the curriculum design problem as finding an ordering over modules as follows:
Given a concept graph $G = \langle V, E \rangle$ with $n > 0$ nodes, u, v in V s.t u is a prerequisite of v and m targeted *learning units* L, where $m \leq n$, generate a list of learning modules $\mathcal{L} = \langle L_1, L_2, , \ldots, L_m \rangle$ that minimizes:

$$\begin{aligned} &\sum_{\substack{\text{Index}(u)<\text{Index}(v) \\ (u,v)\in E}} \text{Distance}(v,u) * \lambda_l \quad + \\ &\sum_{\substack{\text{Index}(u)<\text{Index}(v) \\ (u,v)\in E}} \text{Distance}(u,v) * \lambda_p \quad + \\ &\sum_{\substack{\text{Index}(u)=\text{Index}(v) \\ (u,v)\notin E}} \lambda \end{aligned} \tag{13}$$

where
$\text{Distance}(v, u) = \text{Index}(v) - \text{Index}(u)$ and
$\text{Index}(v)$ is the index of the first learning module in which concept v is covered
λ_l is a penalty associated with violating locality of reference,
λ_p is a penalty associated with violating a prerequisite constraint, and
λ is a penalty associated with covering two unrelated concepts in the same module.
Any of the penalty parameters can be set to 0 if the corresponding constraint is not to be enforced.
We note that this is an NP-hard problem which we can prove by a polynomial-time reducibility to the Minimum Linear Arrangement problem which is known to be NP-hard. In the Minimum Linear Arrangement problem, the goal is to order the nodes of a graph such the total weight (cost) of the edges in the arrangement is minimized. NP-hardness of Minimum Linear Arrangement is proven in [43]. Future work includes generating and evaluating curricula using this approach.

7.2 From Analysis to Evaluation: Criteria and Metrics

Educational curricula are evaluated according to different criteria [36]. We focus on instrinsic criteria that are purely content-based.

7.2.1 Outcome Coverage

A curriculum should have explicit and clearly stated educational objectives and learning outcomes. Educational objectives correspond to general desired properties enjoyed by the learner on the long term. They are usually broad and characterize life-long and self-directed learning pursuit. The area of specialization or focus is one important aspect of the educational objective. Learning outcomes define specific knowledge and skills expected from learners by the time the program is completed. A curriculm, as whole, should cover the intended learning outcomes and contribute towards the educational objectives. Each unit in the curriculum should cover a nonempty subset of the learning or educational objectives (*alignment* of content to outcomes and objectives). Evaluating a curriculum according to coverage of objectives and outcomes is a fitness-*for*-purpose type of evaluation; i.e., *given* a set of intended learning outcomes, the curriculum is evaluated for coverage of these outcomes. We propose hard and soft matching methods to evaluate the set cover of the outcomes by modules in the curricula and analyze the contribution of each module in coverage of stated outcomes.

7.2.2 Overlap

Overlap is measured in terms of similarity and common concepts between courses/modules. While too much overlap indicates redundant content and leads to inefficient use of the learner (and institution) resources, some overlap contributes to coherence and helps to reinforce important concepts. We propose to apply standard document and text similarity approaches [28, 8, 27, 16] to evaluate overlap between different modules in the curriculum.

7.2.3 Vertical and Horizontal Coherence

Vertical coherence refers to coherence observed between units in successive parts of the curriculum. For a program consisting of several parallel paths, vertical coherence is coherence observed in each of the paths. Horizontal coherence refers to coverage similarity and common core between alternative paths in a curriculum plan.

7.2.4 Temporal Stability of Core

Curricula undergo content drift due to shifts in content prioritization and changing research trends. However, it is important to maintain a common core across different offerings of the same module across time. Due to the lack of historical course data, this important aspect of educational content analysis is exlcuded from the thesis. However, for the analysis/evaluation part, given course data across time, we can use tools for modeling temporal dynamics of content to detect longitudinal drift in curricula coverage.

Since the above evaluation is based on analysis of content which is relative, we propose to benchmark against gold sequences given in accredited programs using both similarity of outcome list, module contents, coherence and Kendall Tau score of the analyzed curriculum and the accredited program sequence as a gold standard.

7.2.5 Case Study

As a case study, we evaluate the overlap between the set of courses in the LTI program as of 2016. Information about courses was collected from the LTI online course catalogue. Additionally, course descriptions and detailed syllabi were collected from the different course webpages. Course descriptions vary in length, amount of details, and hosting platform. Since this may introduce a length bias, we base the overlap computation in types not tokens.
The collected syllabi vary in length between 199 concepts (as maximum corresponding to 10-701) and 17 concepts (minimum corresponding to 11-727) with a mean of 77 concepts and standard deviation of 48. The distribution of course size in terms of concepts appears to follow a power law distribution as shown in Figure 10.

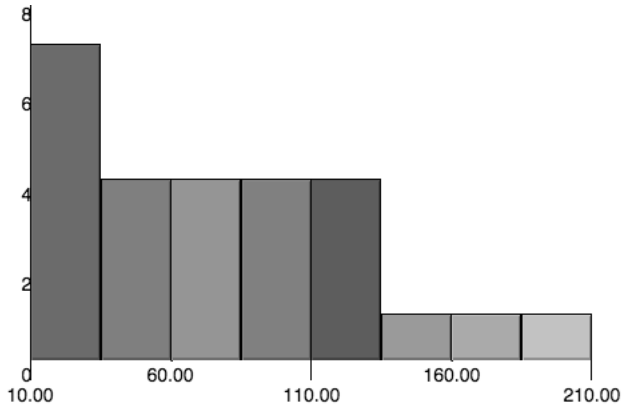

Figure 10: Distrbution of Course Size in Concepts (interval size = 25)

Excluding seminar, lab courses, and research units, the remaining courses are included in the analysis in addition to 10-701. The Stanford parser was used to extract base noun phrases from the dataset. tf.idf was used to filter out irrelevant content. The final set contains 1640 types. Table 12 shows the most frequent concepts and their corresponding frequencies (left top to bottom then right).

type	type
hidden markov models/hmm	evaluation
design	expectation maximization/em
data	inference
machine learning	information retrieval
theory	linguistics
language technologies	models
machine translation	semantics
algorithms	statistics
applications	syntax
programming	classification
analysis	computer science
clustering	dynamic programming
implementation	nlp
language	part-of-speech tagging
speech	question answering
text	regularization
computational linguistics	training

Table 12: Frequent Concepts in LTI Courses

The largest overlap, in terms of the absolute size of common concepts/topics, was found between the following pairs of courses. Pairs in the list are ordered in decreasing order of the Szymkiewicz-Simpson Coefficient which is a similarity measure defined as the size of the intersection of two sets divided by the smaller of the sizes of the two sets. This measure is preferred since it adjusts for length bias.

Language and Statistics	Language and Statistics II
Speech Recognition & Understanding	Language and Statistics
Machine Translation	Language and Statistics II
Machine Learning	Structured Prediction
Machine Translation	Structured Prediction
Natural Language Processing	Speech Recognition
Natural Language Processing	Algorithms for NLP
Machine Learning	Language and Statistics

Table 13: Course pairs with largest overlap

The following is the list of top 10 courses with the largest number of overlapping courses.

course	# overlapping courses
Algorithms for NLP	25
Language and Statistics II	24
Search Engines	23
Grammars and Lexicons	23
Machine Translation	23
Speech Recognition & Understanding	23
Language and Statistics	23
Natural Language Processing	22
Summarization and Personal Information Management Systems	22
Grammar Formalisms	21

Table 14: Courses with the largest number of overlapping courses

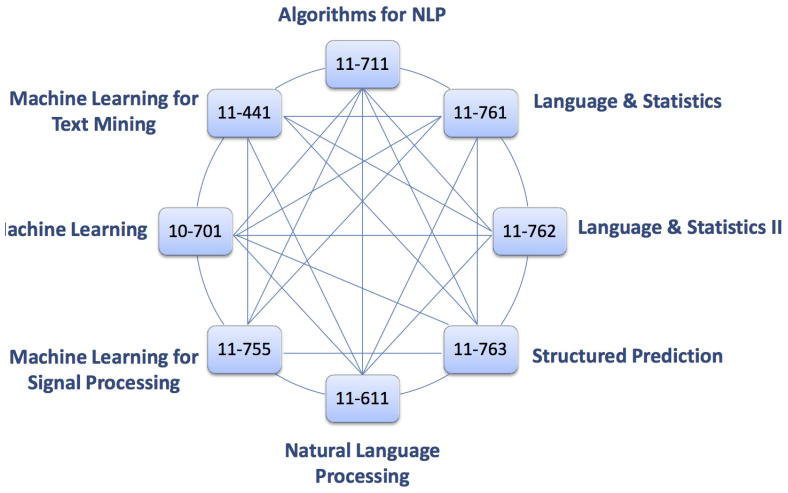

Figure 11: Largest clique of overlapping courses

There's a significant variance in overlap patterns found among the courses. More overlap is observed among courses that belong to the same focus area. Total overlap size is highly correlated with course size in concepts (correlation coefficient = 0.70). Course concept size is also highly correlated with the average overlap size (correlation coefficient = 0.67). Other factors that correlate with overlap (correlation coefficient > 0.5) are 1. existence of prerequisite relation and 2. whether the courses are taught by the same (group of) faculty member(s). The largest clique of overlapping courses is given in Figure 11.

Future work includes building a curriculum evaluation tool based on the pipeline of tools developed for this task from data featurization and concept extraction to overlap and coherence computation and prerequisite prediction methods developed in this thesis. There is a need in the application domain for an automated tool for content analysis, summarization, and evaluation. Currently, internal program review is carried out by humans in a *large bias* setting where a small group of domain experts analyze content for overlap and prerequisites. External review; e.g., in program accreditation, is also carried out by human experts in a *large variance* setting where different teams evaluate different programs. With large amounts of content data, a broad and accurate view of educational content is beyond the resources typically allocated for the task. An automated tool that allows efficient processing and evaluation of large amounts of educational content will improve consistency, reduce bias, and help focus the efforts of domain exports on subtle and more critical aspects of curriculum evaluation.

References

[1] Accreditation Board for Engineering and Technology. ABET accredited programs. `http://main.abet.org/aps/accreditedprogramsearch.aspx`.

[2] ACM IEEE-CS computer science curriculum. `http://www.acm.org/education/CS2013-final-report.pdf`.

[3] R. Agrawal, B. Golshan, and E. E. Papalexakis. Toward data-driven design of educational courses: A feasibility study. In *Proceedings of the 9th International Conference on Educational Data Mining, EDM 2016, Raleigh, North Carolina, USA, June 29 - July 2, 2016*, page 6, 2016.

[4] R. Al-Rfou, B. Perozzi, and S. Skiena. Polyglot: Distributed word representations for multilingual NLP. *CoRR*, abs/1307.1662, 2013.

[5] R. Antonova, J. Runde, M. H. Lee, and E. Brunskill. Automatically learning to teach to the learning objectives. In *Proceedings of the Third (2016) ACM Conference on Learning @ Scale*, L@S '16, pages 317–320, New York, NY, USA, 2016. ACM.

[6] R. Baker. Data mining for education. In P. Peterson, E. Baker, and B. McGaw, editors, *International Encyclopedia of Education*. Elsevier, Oxford, third edition edition, 2010.

[7] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278, Dec 1994.

[8] C. Corley and R. Mihalcea. Measuring the semantic similarity of texts. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, EMSEE '05, pages 13–18, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[9] Coursera. `https://www.coursera.org/`.

[10] Coursera Catalog API. `https://tech.coursera.org/app-platform/catalog/`.

[11] B. Dhingra, H. Liu, R. Salakhutdinov, and W. W. Cohen. A comparative study of word embeddings for reading comprehension. *CoRR*, abs/1703.00993, 2017.

[12] S. Doroudi, K. Holstein, V. Aleven, and E. Brunskill. Sequence matters, but how exactly? a method for evaluating activity sequences from data, 2016.

[13] edX. `https://www.edx.org/`.

[14] M. Faruqui, J. Dodge, S. K. Jauhar, C. Dyer, E. Hovy, and N. A. Smith. Retrofitting word vectors to semantic lexicons. In *Proceedings of NAACL*, 2015.

[15] F. Fouss, A. Pirotte, J.-M. Renders, and M. Saerens. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Trans. on Knowl. and Data Eng.*, 19(3):355–369, Mar. 2007.

[16] W. H. Gomaa and A. A. Fahmy. Article: A survey of text similarity approaches. *International Journal of Computer Applications*, 68(13):13–18, April 2013.

[17] W. S. Gray and B. E. Leary. What makes a book readable. 1935.

[18] R. M. Idrus. *Technogogy: A Convergence of Pedagogy, Technology and Content in Distance Education*. School of Distance Education, Universiti Sains Malaysia, 2007.

[19] R. M. Idrus and K. McComas. Technogogy: Facilitating the transformation of learning. *International Journal of the Computer, the Internet and Management*, 14(SP1):5.1–5.9, 8 2006. An optional note.

[20] B. F. Jones. The burden of knowledge and the 'death of the renaissance man': Is innovation getting harder? Working Paper 11360, National Bureau of Economic Research, May 2005.

[21] Y. Kim. Convolutional neural networks for sentence classification. *CoRR*, abs/1408.5882, 2014.

[22] D. Koutra, U. Kang, J. Vreeken, and C. Faloutsos. Vog: Summarizing and understanding large graphs. *CoRR*, abs/1406.3411, 2014.

[23] Learning analytics. `https://tekri.athabascau.ca/analytics/`.

[24] H. Liu and P. Singh. Conceptnet — a practical commonsense reasoning toolkit. *BT Technology Journal*, 22(4):211–226, Oct. 2004.

[25] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. The Stanford parser. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, 2014.

[26] MetaAcademy project. `https://metacademy.org/`.

[27] D. Metzler, S. Dumais, and C. Meek. Similarity measures for short segments of text. In *Proceedings of the 29th European Conference on IR Research*, ECIR'07, pages 16–27, Berlin, Heidelberg, 2007. Springer-Verlag.

[28] M.K.Vijaymeena and K.Kavitha. A survey on similarity measures in text mining. *Machine Learning and Applications: An International Journal*, 3(2):19–28, 3 2016.

[29] S. K. Mohamad and Z. Tasir. Educational data mining: A review. *Procedia - Social and Behavioral Sciences*, 97:320 – 324, 2013. The 9th International Conference on Cognitive Science.

[30] F. Paas, A. Renkl, and J. Sweller. Cognitive load theory and instructional design: Recent developments. 38:1–4, 06 2010.

[31] L. Pan, C. Li, J. Li, and J. Tang. Prerequisite relation learning for concepts in moocs. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1447–1456. Association for Computational Linguistics, 2017.

[32] A. Peña Ayala. Review: Educational data mining: A survey and a data mining-based analysis of recent works. *Expert Syst. Appl.*, 41(4):1432–1462, Mar. 2014.

[33] A. Pea-Ayala. *Educational Data Mining: Applications and Trends*. Springer Publishing Company, Incorporated, 2013.

[34] J. Pennington, R. Socher, and C. D. Manning. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[35] E. Pollock, P. Chandler, and J. Sweller. Assimilating complex information. *Learning and Instruction*, 12(1):61 – 86, 2002.

[36] A. Porter. Curriculum assessment. In J. L. Green, G. Camilli, and P. B. Elmore, editors, *Complementary methods for research in education*. American Educational Research Association, Washington, DC, 2004.

[37] A. N. Rafferty, E. Brunskill, T. L. Griffiths, and P. Shafto. Faster teaching via pomdp planning. *Cognitive Science*, 2015.

[38] C. Romero and S. Ventura. Educational data mining: A survey from 1995 to 2005. *Expert Systems with Applications*, 33(1):135 – 146, 2007.

[39] C. Romero and S. Ventura. Educational data mining: A review of the state of the art. *Trans. Sys. Man Cyber Part C*, 40(6):601–618, Nov. 2010.

[40] R. Rosenfeld and P. Clarkson. The CMU-Cambridge statistical language modeling toolkit v2. `http://svr-www.eng.cam.ac.uk/~prc14/toolkit.html`.

[41] J. M. Rouly, H. Rangwala, and A. Johri. What are we teaching?: Automated evaluation of cs curricula content using topic modeling. In *Proceedings of the Eleventh Annual International Conference on International Computing Education Research*, ICER '15, pages 189–197, New York, NY, USA, 2015. ACM.

[42] D. Shahaf and C. Guestrin. Connecting the dots between news articles. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 623–632, New York, NY, USA, 2010. ACM.

[43] J. P. I. Silvestre, J. G. Salgado, and D. D. L. I. Sistemes. Approximation heuristics and benchmarkings for the minla problem, 1998.

[44] K. Toutanova, D. Klein, C. Manning, and Y. Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *In Proceedings of HLT-NAACL*, pages 252–259, 2003.

[45] Udacity. `https://www.udacity.com/`.

[46] Y. Yang, H. Liu, J. Carbonell, and W. Ma. Concept graph learning from educational data. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM '15, pages 159–168, New York, NY, USA, 2015. ACM.

[47] H. Yannakoudakis and T. Briscoe. Modeling coherence in ESOL learner texts. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 33–43, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[48] M. V. Yudelson, K. R. Koedinger, and G. J. Gordon. Individualized bayesian knowledge tracing models. In H. C. Lane, K. Yacef, J. Mostow, and P. Pavlik, editors, *Artificial Intelligence in Education: 16th International Conference, AIED 2013, Memphis, TN, USA, July 9-13, 2013. Proceedings*, pages 171–180, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

Appendix A: Course Corpus Common Words

A List of common words and phrases in the course corpus:

course
understanding
learning
human
world
background
used
development
class
part
issues
global
any
students
language
business
theory
apply
I
problems
better
challenges
practice
effective
life
management
level
education
complex
improve
project
introductory
curiosity
enthusiasm
individuals
postgraduates
type
work
willingness
learn
knowledge
social
explore
people
first
important
focus
teaching
experience
able
modern
develop
study
cover
techniques
way
online
provides
teachers
scientific
public
time
discuss
know
ways
only
technology
part
participants
practical
concurrent
experience
familiarity
href
pre-requisites
track
semester
basic
new
understand
including
provide
skills
us
work
need
two
introduction
key
ideas
topic
no
questions
interest
methods
courses
fundamental
around
university
role
various
based
student
look
strategies
school
examine
approach
completion
enrollment
fundamentals
materials
proficiency
year
requirement

Appendix B: Course Corpus Common Phrases

academic_background
advanced_undergraduate
appropriate_background
introductory_course
main_prerequisite
background_knowledge
basic_background
basic_familiarity
basic_knowledge
basic_understanding
capstone_project
college-level_experience
concepts_track_students
course_content
earlier_course
elementary_knowledge
entry_requirements
first_course
first_parts
final_project
formal_pre-requisites
formal_prerequisites
formal_background_knowledge
fundamental_curiosity
general_background
general_interest
graduate_students
graduate_level_course
high_school_level
homework_assignments
learning_tracks
only_background
only_requirement
only_background
open_mind
particular_background
previous_knowledge
previous_exposure
previous_training
previous_course
previous_coursework
prior_background
prior_coursework
prior_experience
prior_exposure
prior_knowledge
prior_training
prior_specialized_background_knowledge
preliminary_knowledge
primary_audiences

recommended_background_content
second_course
solid_course
specific_background
specific_prerequisite_knowledge
specific_prerequisites
special_background
specific_single_textbook
strong_interest
sophomore/junior-level_undergraduate_students
undergraduate_students
technical_background
tough_issue
thorough_understanding
rigorous_course
rudimentary_understanding