Evaluating sources of course information and models of representation on a variety of institutional prediction tasks

Weijie Jiang
University of California, Berkeley
jiangwj@berkeley.edu

Zachary A. Pardos
University of California, Berkeley
zp@berkeley.edu

ABSTRACT

Data mining of course enrollment and course description records has soared as institutions of higher education begin tapping into the value of these data for academic and internal research purposes. This has led to a more than doubling of papers on course prediction tasks every year. The papers often center around a single prediction task and introduce a single novel modeling approach utilizing one or two data sources. In this paper, we provide the most comprehensive evaluation to date of data sources, models, and their performance on downstream prediction tasks. We separately incorporate syllabus, catalog description, and enrollment history data to represent courses using graph embedding, course2vec (i.e., skip-gram), and classic bag-of-words models. We evaluate these representations on the tasks of predicting course prerequisites, credit equivalencies, student next semester enrollments, and student course grades. Most notably, our results show that syllabi bag-of-words representations performed better than course descriptions in predicting prerequisite relationships, though enrollment-based graph embeddings performed substantially better still. Course descriptions provided the highest single representation accuracy in predicting course similarity, with descriptions, syllabi, and course2vec combined representations providing the highest ensembled accuracy on this task.

Keywords

Higher education, course recommendation, course2vec, prerequisites, enrollment histories, syllabus, network embedding, grade prediction, institutional analytics.

1. INTRODUCTION

Data from institutions of higher education are quickly coming into focus for educational data mining and learning analytics communities as the utility of these data start to become clear and attention begins to shift from the informal learning context of free online courses to the higher stakes context of degree granting institutions and their students. Educational Data Mining (EDM) plays an important role in the developing stages of methodological adaptation to a domain by evaluating new sources of data for their utility in existing models and tasks and updating the utility of existing data as models and tasks evolve. Recently, EDM has seen a more than doubling year-to-year in papers focused on prediction with large institutional enrollment sets from the formal higher education context, with a single paper on the topic in 2017 [38], two in 2018 [12, 6], and five in 2019 [29, 36, 19, 37, 16], though early pioneering work on predicting academic outcomes date back to the first EDM conference [39, 2].

In this paper, we summarize and evaluate this quickly developing domain across three dimensions: sources of institutional data, models for representing students and courses, and the performance of the former two categories on institutionally relevant prediction tasks. As academic researchers and practitioners know, not all sources of data are always available and different costs are associated with obtaining a new source. Similarly, when it comes to modeling, different personnel and computational costs are associated with applying models depending on their complexity and recency of introduction. We provide the most comprehensive evaluation to date of the performance of different combinations of data and models on common institutional tasks emerging in the literature so that the costs and benefits of each, in our setting, can be quickly apprised. In addition to evaluating previously introduced approaches and data, we introduce large scale syllabus data as a novel source of information about courses and a novel application of a nascent graph-embedding approach for representing courses.

2. RELATED WORK

Contemporary approaches to data mining institutional datasets in higher education have distinguished themselves from earlier drop-out detection work [18] in the use enrollment data and adoption of representational methods that factorize, embed, or otherwise vectorize courses into a space. This began with [10] that used matrix factorization applied to student enrollments and observed that the factorization grouped courses and students in semantically meaningful ways. Subsequent research also employed matrix factorization for grade prediction tasks [38, 37]. Neural embedding models followed, with the skip-gram neural network model applied to sequences of course enrollments, an approach coined “Course2vec” [32]. The course embeddings extracted from this model were found to be predictive of on-

Weijie Jiang and Zach Pardos "Evaluating sources of course information and models of representation on a variety of institutional prediction tasks" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 115 - 125

Table 1: Related work on institutional prediction tasks (columns) and sources of data used in the task (rows)

	Grade prediction	Enrollment prediction	Prerequisite prediction	Course similarity
Course grades	[10, 15, 21, 38, 37, 16]	[10, 3, 32, 36]	[22, 11, 21, 16]	[17, 25, 12, 29]
Enrollment histories	[10, 15, 21, 38, 37, 16]	[10, 3, 32, 36, 1]	[22, 11, 21, 16]	[17, 25, 31, 33, 29]
Major declarations	[38, 21, 37]	[32, 36]	[21]	[25]
Catalog descriptions		[32]		[26, 31, 33, 12, 29]

time graduation [25], course similarity within [33] and across institutions [31], and of latent topics of courses [8]. Student course selections have also been posed as a graph, treating courses as nodes and student course selections as strengthening the edges between courses the more frequently they share students in common [15, 16, 1]. The aforementioned approaches all use student course selections, a collaborative signal, to represent a course. Other approaches utilize content data of a course (e.g., catalog description) for representation and for downstream tasks such as course similarity analysis [26, 31, 33, 12, 29] and enrollment prediction [32]. Several papers have collected course ratings for modeling and recommendation [13, 12].

The majority of models in related works have been framed as potentially contributing to a course recommendation system, or already integrated into one. They commonly focused on grade prediction [10, 15, 21, 38, 37, 16] as a necessary first-step towards a preparation, or goal-based [21] recommendation system that could aid students in preparing for difficult courses. In a similar vein, prerequisite course inference has been framed [22, 11, 21, 16] also as a potential means to help guide students towards course taking paths expected to be more successful than others [11, 30]. Table 1 summarizes this body of work in terms of the most common data sources used (i.e., course grades, enrollment histories, major declarations, and catalog descriptions) and most common evaluation tasks (i.e., grade prediction, enrollment prediction, prerequisite prediction, and course similarity) focused on in this paper.

3. DATA SOURCES

In this section, we will describe the three primary sources of data utilized in this paper. First, we will describe the source generally, followed by a paragraph detailing the particulars of the dataset used in our offline evaluation experiments.

3.1 Enrollment histories and grades

A student's transcript is classically a report containing the student's histories of courses taken and the grade achieved in each. Enterprise database systems often store raw forms of these data. It has become more common for institutions to not only store these data in relational form but for their internal offices of institutional analytics to have ready access to them. As the fields of EDM and learning analytics have grown, these data have become more available to faculty to aid scholarly research. We used an anonymised enrollments and grades dataset containing student enrollment histories at a large public university, UC Berkeley, collected from Fall 2008 through Fall 2017. The dataset consists of per-semester (i.e., Fall, Spring, and Summer) class enrollments for 164,196 students (both undergraduates and graduates) with a total of 4.8 million class enrollments. A class enrollment record in the data indicates that the student was still enrolled in the class at the end of the semester. The action of dropping a class is not contained in these data. The median number of classes enrolled by a student in a semester was four. There were 9,478 unique lecture courses from 214 departments hosted in 17 different Divisions of 6 different Colleges. Course meta-information was also included in these data and contained course number, department name, class instructor(s), and room max capacity. In this paper, we only consider lecture courses with at least 20 enrollments total over the 9-year period, resulting in 7,487 courses. Although courses can be categorized as undergraduate courses and graduate courses, undergraduates are allowed to enroll in many of the graduate courses. Enrollment data were sourced from the campus' enterprise data warehouse.

3.2 Course catalog descriptions

A paper catalog use to be the primary way in which students could browse all the course offerings at an institution. Fortunately, this has been superseded by online catalogs, most of which are searchable. The catalog contains course numbers, their hosting department, and typically a paragraph or type description of the course. Our dataset contains the most recent catalog description of every course in our enrollment histories. The average catalog description length was 325 words with 489 courses having exceptionally short descriptions of 10 words or fewer. We sourced these descriptions from the campus Office of the Registrar official API for Course information. These descriptions were pre-processed by (1) removing generic, often-seen sentences across descriptions (2) removing stop words (3) removing punctuation, and (4) word lemmatization and stemming.

3.3 Course syllabi from the Learning Management System

A course syllabus is a detailed, chronological list of subjects and assignments that a course will cover, often with other logistical information about course meeting place and time and grading policies. While the syllabus is perhaps an ideal source of information to utilize for content-based representation of a course, it has been an elusive source to conduct research on. This is because few institutions mandate that instructors make their syllabi public and therefore it is uncommon to have syllabi centrally stored by the institution to subsequently make available to researchers. An additional barrier to research availability is that many institutions view a syllabus as an instructor's intellectual property (IP), and therefore not sharable in original form without permission. Our study introduces syllabus data into contemporary predictive models and tasks, but with a caveat that maintains instructor control over the original intellectual property.

The university from which our syllabus data come from considers syllabi to be instructor IP and does not collect them centrally. However, a common place in which instructors often place their syllabi is the ”Syllabus” page of the campus Learning Management System (LMS). We worked with the campus technology services organization in charge of the LMS to extract all text from the Syllabus pages of all courses. Sometimes this page would contain only a link to the pdf of a syllabus, in which case that link was downloaded and parsed to text. To abide by the IP restrictions around course syllabi and respect instructor ownership of them, a workaround was arranged. Only the technology services would have access to the cleanly parsed data from the LMS. They would then pre-process the syllabus themselves, similar to how we pre-processed catalog descriptions, parsing out html, converting it into bag-of-words (BOW) form. This form would thereby make the syllabus unusable as an instructional object but potentially usable by an algorithm attempting to extract information for institutional prediction tasks. It was also agreed that the BOW we received would not be made public and these data could be revoked at any time. There were 3,645 unique courses that contained HTML on the LMS Syllabus page, not including a link to a file. There were 2,712 courses that contained a link to a file, with some courses having both. The total number of courses with some amount of syllabus data was 4,017 with a combined vocabulary of 17,194 unique words.

4. REPRESENTATION MODELS

We choose four approaches of increasing complexity for representing courses. These four reflect the most common paradigms of modeling found in our literature review. The simplest is a content-based bag-of-words representation of the course. The BOW approach could be applied to the catalog description or syllabus of a course, where available. Next is the use of a recently published variant on Course2vec called multifactor Course2vec, which applies a skip-gram to sequences of course enrollments. In addition to embedding courses, multifactor Course2vec also embeds the instructor of the course and the course's department, both presented to the model in the form of a one-hot encoding. Multifactor Course2vec has been shown to perform better on course similarity tasks than the original Course2vec [33], in theory because it separates out factors, such as instructor and department, allowing the course embedding to more purely represent the content. Long Short-Term Memory models are the third model used to embed courses, followed by a recently introduced network embedding technique.

A summary of the approaches used is visually illustrated in Figure 2. The various types of information these methods leveraged are summarized in Table 2.

Table 2: Summary of representative learning methods for courses

	catalog descriptions	course syllabus	course meta-information	enrollment histories	course grades	model type
bag-of-words	✓	✓				static
multi-c2v			✓	✓		dynamic
LSTM				✓		dynamic
sc-AMHEN				✓	✓	static

4.1 Bag-of-words

The basic representation mode of bag-of-words was proposed by information retrieval researchers for text corpora. It is a model that reduces each document in a corpus to a vector of real numbers, each of which represents a term, or vocabulary weight. The term weight can be term frequency, a binary value with 1 indicating that the term occurred in the document and 0 indicating that it did not, or a tf-idf scheme[7]. There are two sources of texts that can represent the content of courses: the course catalog descriptions and course syllabi.

4.2 Multifactor Course2vec

The Course2vec model [32] was proposed to learn distributed representations of courses from students' enrollment records throughout semesters by using a notion of an enrollment sequence as a "sentence" and courses within the sequence as "words", borrowing terminology from the natural language domain. For each student, their chronological course enrollment sequence is produced by first sorting by semester then randomly serializing within-semester course order. Each course enrollment sequence is then trained on like a sentence using a skip-gram model.

More features of courses (e.g., course instructor and department) can be added to the input of the multifactor Course2vec model to enhance the classifier and its representations. The model learns both course and added feature representations by maximizing the objective function over all the students' enrollment sequences and the features of courses, defined as follows.

$$\sum_{s\in S}\sum_{c_i\in s}\sum_{-w<j<w,j\neq 0}\log p(c_{i+j}|c_i,f_{i1},f_{i2},...,f_{ih}) \qquad (1)$$

Probability $p(c_{i+j}|c_i,f_{i1},f_{i2},...,f_{ih})$ of observing a neighboring course c_{i+j} in window size w given the current course c_i and its features $f_{i1},f_{i2},...,f_{ih}$ (e.g., instructors, department) can also be defined via the softmax function,

$$p(c_{i+j}|c_i,f_{i1},f_{i2},...,f_{ih}) = \frac{\exp(\boldsymbol{a}_i^T\boldsymbol{v}'_{i+j})}{\sum_{k=1}^{n}\exp(\boldsymbol{a}_i^T\boldsymbol{v}'_k)} \qquad (2)$$

$$\boldsymbol{a}_i = \boldsymbol{v}_i + \sum_{j=1}^{h}\boldsymbol{W}_{n_j\times v}\boldsymbol{f}_{ij} \qquad (3)$$

where $\boldsymbol{a}_c$ is the vector sum of input course vector representation $\boldsymbol{v}_c$ and all the features vector representations of course c, $\boldsymbol{f}_{ij}$ is the multi-hot input of the j-th feature of course i, and $\boldsymbol{W}_{n_j\times v}$ is the weight matrix for feature j. So by multiplying $\boldsymbol{W}_{n_j\times v}$ and $\boldsymbol{f}_{ij}$, it gets the sum of feature vector representations of the i-th course. The illustration of the model is shown in the multi-course part of Figure 2. $\boldsymbol{v}_i$ is the course representation of course i learned from the model that is used in various down-stream course prediction tasks.

4.3 LSTM-learned Representations

In previous work [32], an LSTM was designed to recommend courses for students to take in the next semester, based on their enrollment histories. The input of the model in each time slice is a multi-hot vector representing the courses taken in the corresponding semester. The weights of the input $\boldsymbol{W}_f$, $\boldsymbol{W}_i$, $\boldsymbol{W}_o$, and $\boldsymbol{W}_c$ learned by the LSTM transferred the

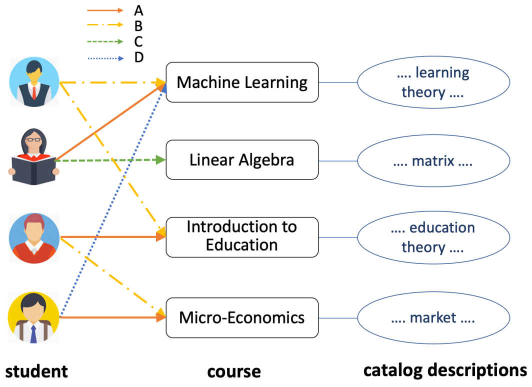


Figure 1: Illustration of the Attributed Multiplex HEterogeneous Network (AMHEN) of Students and Courses.

multi-hot input to the forget gate, input gate, output gate, and the cell in the LSTM cell, respectively. These four sets of weights are combined to form representations of courses that can be used in down-stream prediction tasks.

4.4 Attributed Multiplex Heterogeneous Network Embeddings

Network representation learning (i.e., network embedding), is a promising method to project nodes in a network onto a low-dimensional continuous space while preserving network structure and inherent properties. In terms of the network topology (homogeneous or heterogeneous) and attributed property (with or without attributes), six different types of networks can be categorized, i.e., HOmogeneous Network (HON) [34], Attributed HOmogeneous Network (AHON) [40], HEterogeneous Network (HEN) [9], Attributed HEterogeneous Network (AHEN) [5], Multiplex HEterogeneous Network (MHEN) [24], and Attributed Multiplex HEterogeneous Network (AMHEN) [4]. In the university setting, students and courses can be mapped into a large heterogeneous network, where students and courses are two types of nodes connected by students' enrollments in courses. The proximities between students and courses vary based on the grades (e.g., A, B, C, D, etc.) students received for courses, yielding the network with multiple views, i.e., multiplex heterogeneous network. Furthermore, if we incorporate the attributes of students and nodes (e.g., course catalog descriptions), the network will turn to an Attributed Multiplex HEterogeneous Network (AMHEN), which is illustrated in Figure 1. Because students may receive different grades for the courses they enrolled, we consider different grades as different edge types between students and courses.

Definition 1. *(Attributed Multiplex Heterogeneous Network): An attributed multiplex heterogeneous network is a network $G = (\mathcal{V}, \mathcal{E}, \mathcal{A})$, $\mathcal{E} = \cup_{r\in\mathcal{R}} \mathcal{E}_r$,where $\mathcal{E}_r$ consists of all edges with edge type $r \in \mathcal{R}$, and $|\mathcal{R}| > 1$. We separate the network for every edge type $r \in \mathcal{R}$ as $G_r = (\mathcal{V}, \mathcal{E}_r, \mathcal{A})$. Each node $v_i \in \mathcal{V}$ is associated with some types of feature vectors. $A = \{x_i|v_i \in \mathcal{V}\}$ is the set of node features for all nodes, where x_i is the associated node feature of node v_i.*

In the student-course attributed multiplex heterogeneous network we described above, $\mathcal{V} = (\mathcal{C}, \mathcal{S})$, where each node $c \in \mathcal{C}$ represents a course in the course set $\mathcal{C}$ and each node $s \in \mathcal{S}$ represents a student in the student set $\mathcal{S}$. $\mathcal{R}$ refers to all the edge types in the student-course attributed multiplex heterogeneous network, i.e., grade types. As students have enrollment and grade histories of multiple courses, we consider student embeddings as a state of their course knowledge. Different grade types mirror different levels of course knowledge, thus should be represented as different embeddings.

Given the above definitions and descriptions, we can formally define our problem for representation learning on the student-course AMHEN.

Problem 1. *(Student-Course AMHEN Embedding). Given a Student-Course AMHEN $G = (\mathcal{C}, \mathcal{S}, \mathcal{E}, \mathcal{A})$, the problem of Student-Course AMHEN embedding is to give a unified low-dimensional space representation of each student node $s \in \mathcal{S}$ and each course node $c \in \mathcal{C}$ on every grade type r.The goal is to find a function $g : \mathcal{S} \to \mathbb{R}^d$ and a function $f_r : \mathcal{C} \to \mathbb{R}^d$ for every grade (edge) type r, where $d \ll |C|$ $(d \ll |S|)$.*

4.4.1 Student and Course Representations

In this section, we detail our adaptation of the AMHEN framework[4] to the student-course scenario to learn graph-based student and course representations. We split the overall course embedding on each course type r into three parts: base embedding $\boldsymbol{b_c}$, grade embedding $\boldsymbol{g}$, and attribute embedding $\boldsymbol{u}$, and split the overall student embedding into two parts: base embedding $\boldsymbol{b_s}$, and individual embedding $\boldsymbol{p}$.

The base embedding of course node c_i, i.e., $\boldsymbol{b_{c}}_i$, is shared between different grade types. We define $\boldsymbol{b_{c}}_i$ as a parameterized function of c_i's attributes $\boldsymbol{x}_i \in \mathbb{R}^x$ as:

$$\boldsymbol{b_{c}}_i = h(\boldsymbol{x}_i) \tag{4}$$

where h is a transformation function, such as a multi-layer perceptron. The attribute embedding of course node c_i, i.e, $\boldsymbol{u}_i$, is defined as:

$$\boldsymbol{u}_i = D^T \boldsymbol{x}_i \tag{5}$$

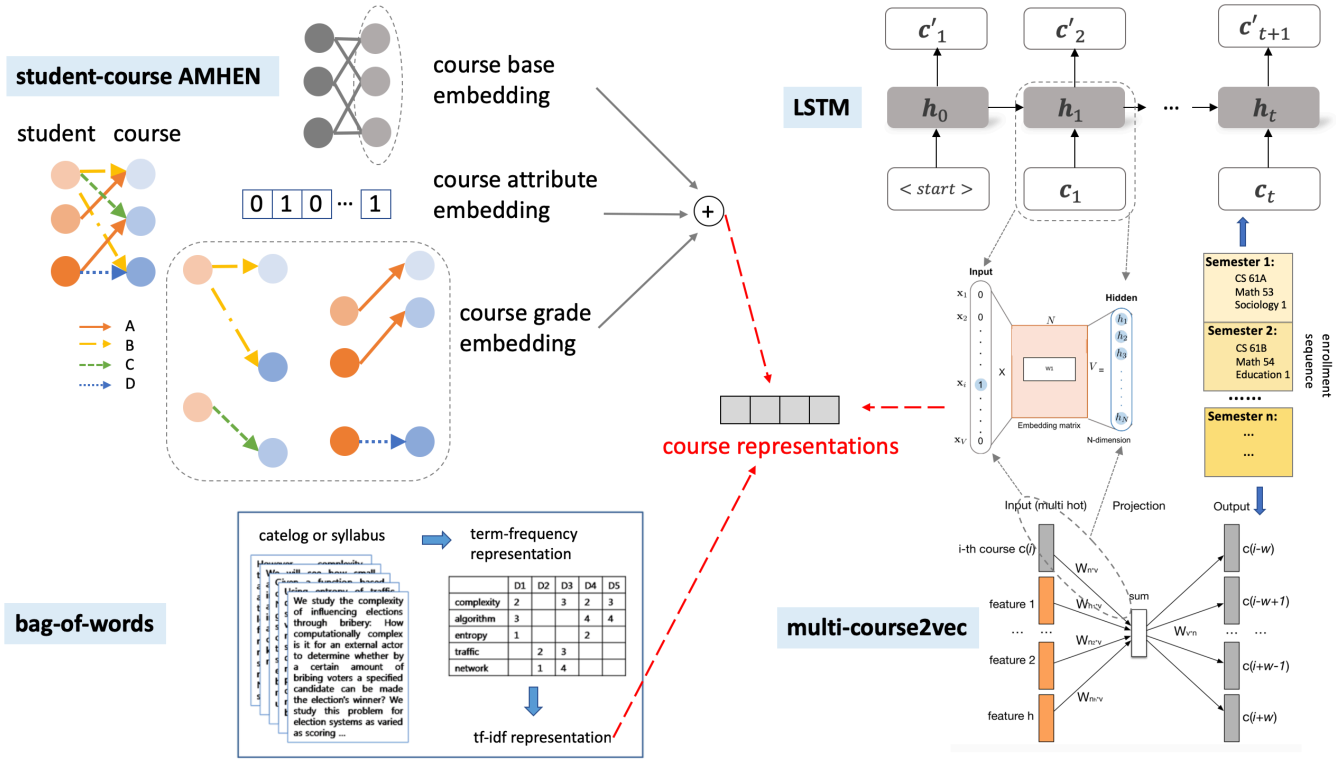
Given that in the Student-Course AMHEN, the neighbors of a course are all students while the neighbors of students are all courses, the k-th level[1] of grade embedding $\boldsymbol{g}_{ir}^{(k)} \in \mathbb{R}^d$, $(1 \leq k \leq K)$ of course node c_i on grade type r is aggregated from individual embeddings of students that are c_i's neighbors, which means these students all received grade type r for course c_i.

$$\boldsymbol{g}_{ir}^{(k)} = mean(\{\boldsymbol{p}_j^{(k-1)}, \forall p_j \in \mathcal{N}_i\}) \tag{6}$$

Similarly, the k-th level of individual embedding $\boldsymbol{p}_i^{(k)} \in \mathbb{R}^d$, $(1 \leq k \leq K)$ of a student node s_i is aggregated from grade embeddings of courses that are s_i's neighbors, which demonstrates a student's representation is derived from the grade histories of his/her enrolled courses.

$$\boldsymbol{p}_i^{(k)} = mean(\{\boldsymbol{g}_{jr}^{(k-1)}, \forall c_j \in \mathcal{N}_{ir}\}) \tag{7}$$

[1] By level we mean iteration, i.e., the embedding is updated after each parameters update process.



We study the complexity of influencing elections through bribery: How computationally complex is it for an external actor to determine whether by a certain amount of bribing voters a specified candidate can be made the election's winner? We study this problem for election systems as varied as scoring ...

	D1	D2	D3	D4	D5
complexity	2		3	2	3
algorithm	3			4	4
entropy	1			2	
traffic		2	3		
network		1	4		

Figure 2: Visual summary of representation learning methods

We denote the k-th level grade embedding $\boldsymbol{g}_{ir}^{(k)}$ as grade embedding $\boldsymbol{g}_{ir}$, and concatenate all the grade embeddings for course node c_i as $\boldsymbol{G_i} \in \mathbb{R}^{d\times m}$, where d is the dimension of grade embeddings and m is the number of grade types.

$$\boldsymbol{G_i} = (\boldsymbol{g}_{i1}, \boldsymbol{g}_{i2}, ..., \boldsymbol{g}_{im}) \tag{8}$$

We use self-attention mechanism[23] to compute the coefficients $\boldsymbol{a}_{ir} \in \mathbb{R}^m$ of linear combination of vectors in $\boldsymbol{G_i}$ on edge type r as:

$$\boldsymbol{a}_{ir} = \mathrm{softmax}(\boldsymbol{w}_r^T \tanh(\boldsymbol{W_r G_i}))^T \tag{9}$$

where $\boldsymbol{w}_r \in \mathbb{R}^{d_a}$ and $\boldsymbol{W_r} \in \mathbb{R}^{d_a\times d}$ are trainable parameters for grade type r. Thus, the overall embedding of course node c_i for grade type r is:

$$\boldsymbol{c}_{ir} = \alpha_c h(\boldsymbol{x}_i) + \boldsymbol{M}_r^T \boldsymbol{G_i a}_{ir} + \beta_c \boldsymbol{D}^T \boldsymbol{x}_i \tag{10}$$

where $\boldsymbol{M}_r \in \mathbb{R}^{d\times n}$ and $\boldsymbol{D} \in \mathbb{R}^{x\times n}$ are trainable transformation matrix. α_c and β_c are two coefficients adjusting the weights of the three embeddings of courses, which can also be trainable.

The overall embedding of student node s_i is:

$$\boldsymbol{s}_i = \alpha_s \boldsymbol{b_s} + \boldsymbol{N}^T \boldsymbol{p}_i \tag{11}$$

where α_s is a trainable coefficient adjusting the weights of the two embeddings of students, and $\boldsymbol{N} \in \mathbb{R}^{d\times n}$ is a trainable transformation matrix for the individual embeddings of students.

4.4.2 Model Optimization

Having the student and course representations constructed, we discuss how to generate the training data and learn the student and course embeddings. We first separate the whole network by edge(grade) type, then given a view (grade type) r of the network, i.e., $\boldsymbol{G}_r = (\mathcal{C}, \mathcal{S}, \mathcal{E}_r, \mathcal{A})$, we use meta-path-based random walk[9] to generate node sequences. There are two meta-path schema in the student-course AMHEN, i.e., $student - course - student$ or $course - student - course$. Finally, we apply a skip-gram [27, 28] over the node sequences to learn embeddings. The meta-path-based random walk strategy ensures that the semantic relationships between student nodes and course nodes with different grade types can be properly incorporated into the skip-gram model [9]. For a training pair (c_i, s_j) with grade type r, our objective is to maximize the probability:

$$P(s_j|c_i, r) = \frac{\exp(\boldsymbol{c}_{ir}^T \boldsymbol{s}'_j)}{\sum_{s_k \in \mathcal{S}} \exp(\boldsymbol{c}_{ir}^T \boldsymbol{s}'_k)} \tag{12}$$

where $\boldsymbol{s}'_k$ is the context embedding of student node s_k. For a training pair (s_i, c_j) with grade type r, our objective is to maximize the probability:

$$P(c_j|s_i, r) = \frac{\exp(\boldsymbol{s}_i^T \boldsymbol{c}'_{jr})}{\sum_{c_k \in \mathcal{C}} \exp(\boldsymbol{s}_i^T \boldsymbol{c}'_{kr})} \tag{13}$$

where $\boldsymbol{c}'_{kr}$ is the context embedding of course node c_k with grade type r. Finally, we use heterogeneous negative sampling to approximate the objective function $-\log P(s_j|c_i, r)$ for node pair (c_i, s_j) as

$$loss(c_i, s_j, r) = -\log\sigma(\boldsymbol{c}_{ir}^T \boldsymbol{s}'_j) - \sum_{l=1}^{L} \mathbb{E}_{s_k \sim P(s_k)} [\log\sigma(-\boldsymbol{c}_{ir}^T \boldsymbol{s}'_k))] \tag{14}$$

and the objective function $-\log P(c_j|s_i, r)$ for node pair (s_i, c_j) as:

$$loss(s_i, c_j, r) = -\log\sigma(\boldsymbol{s}_i^T \boldsymbol{c}'_{jr}) - \sum_{l=1}^{L} \mathbb{E}_{c_k \sim P(c_k)}[\log\sigma(-\boldsymbol{s}_i^T \boldsymbol{c}'_{kr}))] \tag{15}$$

Here we define $P(s_k) = \frac{f(s_k)^{3/4}}{\sum_{i=1}^{|S|} f(s_i)3/4}$ and $P(c_k) = \frac{f(c_k)^{3/4}}{\sum_{i=1}^{|C|} f(c_i)3/4}$ according to the Skip-gram model[27], where f refers to the frequency of the node in each node type.

After optimizing the model with all the parameters learned, we reform the overall embedding for course i by concatenating its embeddings of all grade types.

$$\boldsymbol{c}_i = (\boldsymbol{c}_{i1}^T, \boldsymbol{c}_{i2}^T, ..., \boldsymbol{c}_{im}^T)^T \tag{16}$$

5. TASKS

In this section, we describe five down-stream institutionally relevant tasks that can be performed by using the course representations constructed by the model approaches introduced in Section 4.

5.1 Course Similarity

An essential way to check the quality and fidelity of the course representations introduced in section 4 is to test whether they contain important features of courses that could differentiate between similar and dissimilar courses. To this end, an equivalency validation set of 1,351 course credit-equivalency pairs maintained by the Office of the Registrar were used for similarity based ground truth. A course is paired with another course in this set if a student can only receive credit for taking one of the courses at the university. For example, an honors and non-honors version of the same course will appear as a pair because faculty have deemed that there is too much overlapping material between the two for a student to receive credit for both.

To evaluate different course representations on the course equivalency validation set, we fixed the first course in each pair and ranked all the other courses according to their cosine similarity to the first course in descending order. We then noted the rank of the expected second course in the pair and describe the performance of each model on all validation pairs in terms of Mean Rank, Median Rank and Recall@10.

5.2 Enrollment Prediction

Enrollment prediction involves predicting the courses a student will enroll in, but not the grade they will receive. For this reason, it is considered a model of behavior, rather than an assessment model. The task could be potentially useful for the purpose of providing a normative course taking signal that could be used to provide a personalized sorting of course results (e.g., showing the courses a student is most likely to take that satisfy a remaining requirement) [32]. The input of the model in each time slice is a multi-hot vector representing the courses taken in the corresponding semester. However, the multi-hot representation has a large dimension of total number of courses and may not encode course features apparent in text descriptions of the course or graph-based methods. Therefore, we also evaluate substituting the multi-hot course input with the sum of pre-trained low-dimensional representations from other models, illustrated in Figure 3. Performance on this task is reported in terms of Recall@ 10 and Mean Reciprocal Rank@10 (MRR@10). MRR evaluates recommender system models that produce a list of ranked items for queries. The reciprocal rank is the “multiplicative inverse” of the rank of the first correct item. MRR is defined as MRR $= \frac{1}{|Q|} \sum_{i=1}^{Q} \frac{1}{\text{rank}_i}$, where rank$_i$ represents the rank of the first correct recommended item for query i. For calculating MRR@10, the only difference is rank$_i$ is reset to 0 if rank$_i > 10$.

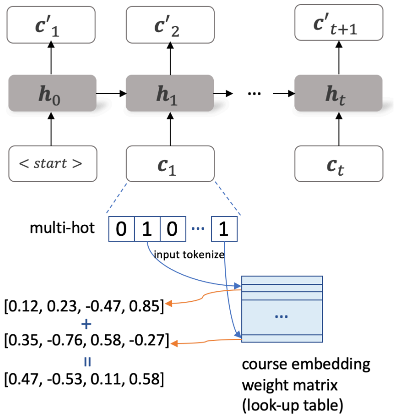


Figure 3: Illustration of the LSTM-based next-course prediction

5.3 Grade Prediction

Grade prediction is the basis for an assessment model that could aid adaptive sequencing of courses to achieve a particular goal. In previous work[21], a modified LSTM was designed to trace students' course knowledge, which predicted students' grades on enrolled courses in each semester. The model gives students the ability to choose their grade goal (A or B) or Pass/No-pass. A masked loss function was designed to enable the output to predict letter grade and Pass/No-pass independently. Two cut-offs (A or B) were also set to separate the letter grades into two levels (e.g., higher and lower than an ’A’). The input of the LSTM grade prediction model is also a multi-hot vector with the position of grades students received for enrolled courses as 1 and other positions as 0. Because there are seven grade types for each course, the dimensions of the model input in each time slice is the number of courses multiplied by seven. As an alternative to the multi-hot input, we also evaluate the performance of the model using the course grade representations learned from the student-course AMHEN model in Section 4.4, which is illustrated in Figure 4, where $\boldsymbol{g}_i$ represents the grades of courses taken in semester i and $\boldsymbol{c}_i$ represents the courses taken in semester i. $\boldsymbol{c}_{i+1}$ is concatenated with $\boldsymbol{g}_i$ to incorporate the impact of the co-enrolling effect of courses in the predicted semester on grade prediction.

In addition, the student-course AMHEN model can also predict the grades of students by calculating the cosine similarities between student embeddings and course embeddings, and then predicting the grades by picking up the grade of each course that is most similar to the target student.

$$g(s_i, c_j) = \arg\max_r cos(\boldsymbol{s}_i, \boldsymbol{c}_{jr}) \tag{17}$$

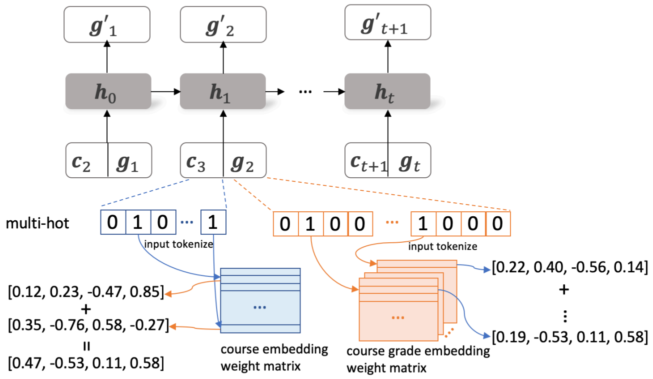


Figure 4: Illustration of the LSTM-based grade prediction

For the model without grade cut-off, there are seven grade types in the student-course AMHEN model representing A, B, C, D, F, Pass, and No-pass. A prediction is considered correct only if it is exactly the grade a student received in the data. For the models with grade cut-off (A or B), we group the letter grades not lower than the cut-off as a grade type, and the letter grades lower than the cut-off as another grade type in the student-course AMHEN model.

Both the enrollment prediction and grade prediction models were trained using a temporal train/test split, with Fall 2008 through Fall 2015 semesters serving as the training set and Spring 2016 as the testing semester.

5.4 Prerequisite prediction

Prerequisite course information is essential to encourage or mandate that students have the necessary foundational experience to be able to learn and succeed in the advanced stages of their degree. We used a set of 2,300 prerequisite course pairs, provided by the UC Berkeley Office of the Registrar, which contains 1,215 target courses, as a source of ground truth to test whether the grade prediction model encodes such prerequisite relationships between courses.

Prerequisite relationships between courses can be inferred by inferencing an LSTM-based grade prediction model as described in [21] and illustrated in Figure 5. Note that, for this evaluation, only one time slice input of the binary-grade (A or lower than A) prediction trained LSTM is needed. We iterate over all the courses with only one-hot embedded in the 'A' position for that course, and feed the input, which is a concatenation of a target course and grade A of the input course, to the LSTM. During the iterations, the input course that boosted the probability of the 'A' position of target course to the largest ten values will be selected as candidate prerequisite courses for the target course. This approach is similar to the prerequisite skill inference conducted with DKT [35], but with a much larger vocabulary and with ground truth prerequisite structure to validate against. As with the other tasks, we also evaluate replacing the input of this model with representations from the student-course AMHEN graph-embedding approach.

A simple multinomial logistic regression can alternatively be used to predict prerequisites courses using any arbitrary vector representation of a course. The input of the multinomial logistic regression during training is the vector representation of the target course, and the output is a multi-hot of the prerequisite courses for the target course. During testing, the output is a probability distribution across all courses where the most probable courses can be taken as the prerequisite predictions of the regression.

We classified all the models for the prerequisite course prediction task into two types, supervised and unsupervised, based on whether the model was learned using the official prerequisite course pairs. For the supervised models (i.e., using the regression), we applied 10-fold cross-validation to the 2,300 prerequisite course pairs. For the unsupervised models (i.e., LSTM-based inferences), described in Section 5.4, the LSTM with standard course multi-hots as input and with graph-based embeddings as input was trained first on the supervised task of predicting course grades, and was then inferenced in an unsupervised manor (i.e., not using any prerequisite ground truth), to predict course prerequisites.

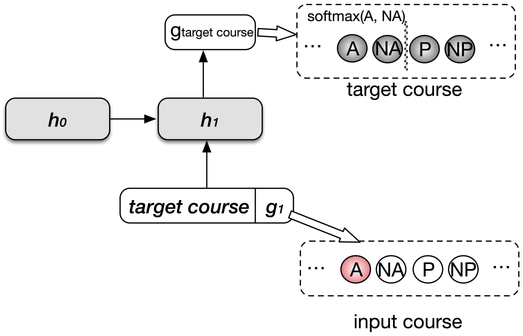


Figure 5: Prerequisite course prediction using LSTM-based grade prediction model[21]

5.5 Average Enrollment Prediction

Do the representations of courses created by various modeling techniques encode course popularity information? To answer this we test the course representations' ability to predict the average enrollment size of each course. The data and models that perform well in this test may be indicative of the data and modeling paradigms that would work well for temporal versions of this model that could anticipate increases in course demand and allow institutions to better plan room and teaching staff allocations.

In order to check whether the different types of course embeddings encode information predictive of the number of enrollments, we use a simple a multi-layer perceptron to predict average enrollment per course using the different types of course embeddings introduced in section 4 as candidate inputs. RMSE is adopted as the error metric.

6. EXPERIMENT RESULTS

We begin this section by reporting a summary of only the best performing model and data source pairs used to construct the input representations for each of our five downstream model predictions tasks. This summarized set of best

Table 3: Evaluation of course representation models on various prediction tasks

Representation created by		**Course similarity prediction**		**Enrollment prediction**		**Grade prediction**	**Prerequisite prediction**		**Avg-enroll-predict**
Model	Data Source(s)	Mean/ Median Rank	Recall@10	Recall@10	MRR@10	Accuracy	Recall@10	Target	RMSE
bag-of-words	catalog	602/**6** [33]	**0.5370**[33]	0.3154	0.5216	-	0.5152	0.5938	42.4781
bag-of-words	syllabus	329/19	0.4270	0.3744	0.5103	-	0.5658	0.6352	48.8965
multi-c2v	enrollments, course meta-information	**224**/15[33]	0.4485[33]	0.3791	0.5576	-	0.6957	0.7733	**42.4780**
LSTM (multi-hot)	enrollments	584/58	0.2924	**0.3967**	**0.5885**	0.6952	0.3048[21]	0.4486[21]	51.4140
sc-AMHEN	enrollments, grades, catalog	288/11	0.4767	0.3882	0.5625	**0.7008**	**0.7192**	**0.8000**	52.3370

results are shown in Table 3. On the task of course similarity, a simple bag-of-words representation of the course catalog description performs best in terms of median rank and Recall @ 10 on our credit-equivalency pairs validation set. Enrollment histories provide the second best performing score using sc-AMHEN network-based embedding, followed by multi-c2v. Scoring similarly to multi-c2v was a simple BOW of the lms-syllabus data. On the task of predicting which courses a student will take next (enrollment prediction), an LSTM with a multi-hot input representation of courses taken in each semester provided the best performance in terms of both metrics. In this task, using pre-trained embeddings from the network-based or multi-c2v approach worked less well than multi-hot, followed by using the content-based representations as inputs, which performed worst. In grade prediction, the network-based method performed slightly better than the previous state-of-the-art LSTM. On the task of prerequisite prediction, the network-based approach performed best in recovering the ground-truth prerequisite relationships found in our institutional data. The multi-c2v approach was not far behind. The content-based and LSTM course representations did not perform nearly as well on this task. Finally, on the task of predicting the average enrollment of a course, multi-c2v provided the lowest RMSE, but with an almost identical score achieved by simple BOW of the course catalog description.

In the subsequent sections we provide a more detailed breakdown of performance of all model and data combinations on the tasks of course similarity, grade prediction, and prerequisite prediction. Results of enrollment prediction and average enrollment prediction are already shown in full in Table 3.

6.1 Course Similarity

The evaluation results on the equivalency validation set of 1,351 course credit-equivalency pairs are shown in Table 4. The bag-of-words representations (Tf-idf) generated from course catalog descriptions achieved better median rank and recall@10 than those generated from the course syllabus data. However, the mean rank of the catalog-based representations is the worst among all the models, which suggests there are many outliers where literal semantic similarity (bag-of-words) is very poor at identifying equivalent pairs. Concatenations of the bag-of-words based methods and course2vec-based method increased the evaluation metrics, especially when the bag-of-words representations of catalog and syllabus were combined with the multi-factor course2vec representations, reaching a mean/median rank of 79/3 and recall@10 of 0.6705, the best among all the models. A Principal Component Analysis (PCA) transformation of the concatenated course vectors from 10,000 to 300 did not diminish the median rank metric, but slightly negatively affected average rank and recall. The course representations learned from the next-course prediction LSTM performed the worst among all the models. Course attribute embeddings sourced from the student-course AMHEN (sc-AMHEN) model, performed second best among all single representation models.

Table 4: Course similarity validation of all the course representations

Model	Mean/Median Rank	Recall @10
catalog	602/6	0.5372
syllabus	329/19	0.4270
course2vec (c2v)	244/21	0.3839
multi-c2v (mc2v)	224/15	0.4485
catalog+mc2v	132/3	0.6435
syllabus+mc2v	109/6	0.5798
catalog+syllabus+mc2v	**79/3**	**0.6705**
catalog+syllabus+mc2v (PCA dim: 300)	177/3	0.6544
LSTM	584/58	0.2924
sc-AMHEN($\boldsymbol{u}$)	288/11	0.4767
sc-AMHEN($\boldsymbol{c}$)	330/27	0.3603

6.2 Grade Prediction

The *accuracy* of the grade predictions generated by the pure student-course AMHEN model (sc-AMHEN($\boldsymbol{s}, \boldsymbol{c}$)), the LSTM model with mult-hot as input (LSTM(multi-hot)), and the LSTM model with course embeddings with different grade types (LSTM($\boldsymbol{u}, \boldsymbol{c}$)) are listed in Table 5. Among the three models, the pure student-course AMHEN model is a kind of static model learned from students' enrollment data with grades and course catalog descriptions, while the two LSTM-based models are dynamic models taking into consideration not only the student enrollment data with grades, but also the sequential informaion (semester order) of the grades of enrolled courses. The grade prediction results show that the graph model, though static, could map the knowledge

Table 5: Grade prediction evaluation (accuracy)

Model	Type	Cut-off	Letter grade	Pass/ No-pass	All
sc-AMHEN $(\boldsymbol{s}, \boldsymbol{c})$	static	-	0.5441	0.7972	0.5976
LSTM (multi-hot)	dynamic	-	0.6382	0.9079	0.6952
LSTM $(\boldsymbol{u}, \boldsymbol{c})$	dynamic	-	**0.6418**	**0.9209**	**0.7008**
sc-AMHEN $(\boldsymbol{s}, \boldsymbol{c})$	static	A	0.5526	0.7791	0.6004
LSTM (multi-hot)	dynamic	A	0.7523	0.8581	0.7633
LSTM $(\boldsymbol{u}, \boldsymbol{c})$	dynamic	A	**0.7571**	**0.9135**	**0.7902**
sc-AMHEN $(\boldsymbol{s}, \boldsymbol{c})$	static	B	0.8299	0.8205	0.8279
LSTM (multi-hot)	dynamic	B	0.8805	0.9178	0.8884
LSTM $(\boldsymbol{u}, \boldsymbol{c})$	dynamic	B	**0.8817**	**0.9185**	**0.8895**

levels of students on the features of courses with different grade types to a certain degree, resulting in prediction accuracies higher than 0.5 for all grade types and higher than 0.6 and 0.8 for binary grades (“not lower than cut-off” v.s.“lower than cut-off”, Pass v.s. No-pass) on average. Furthermore, the sequential information of students’ grades by semesters exhibited substantial importance as the prediction accuracy of the two LSTM-based models manifested superiortiy to the static student-course AMHEN model by a significant margin. Moreover, the course embeddings with different grade types learned from the student-course AMHEN model helped increase the accuracy of grade prediction over the multi-hot vectors as the input of the LSTM. The potential reasons could be the course embeddings with different grade types captured the knowledge relations among grades of a course and the relations among different courses, thus could represent the knowledge of students more accurately than multi-hot, which could not encode any knowledge relations among grades. Although the positive impact of incorporating grade embeddings on grade prediction (improvement at the 0.01 level) are not so salient as the advantage of bringing in sequential information (improvement at the 0.1 level), it is manifested in all the evaluations with different grade types.

6.3 Prerequisite prediction

The evaluation results of prerequisite course prediction are shown in Table 6. The supervised models performed dramatically better in reconstructing the prerequisite pairs. Among all types of course representations, the course embeddings and grade embeddings learned from the student-course AMHEN performed the best, reaching 71.92% of the prerequisite pairs correctly predicted and 80% of all the target courses with at least one of their prerequisite course correctly predicted. For unsupervised models, we found one-hot representation of courses performed better than course and grade embeddings in the prerequisite course inference framework described in Section 5.4.

Table 6: Prerequisite course prediction

Model	Supervised	Pairs (Recall@10)	Target course
LSTM(one-hot)	✗	**0.3048**	**0.4486**
LSTM$(\boldsymbol{u}, \boldsymbol{c})$	✗	0.2423	0.3580
catalog	✓	0.5152	0.5938
syllabus	✓	0.5658	0.6352
mc2v	✓	0.6957	0.7733
sc-AMHEN$(\boldsymbol{u}, \boldsymbol{c})$	✓	**0.7192**	**0.8000**

7. CONCLUSIONS

In this paper, we evaluated the utility of two content-sources of data about courses, catalog descriptions and syllabi, as well as enrollment histories and grades. We paired these sources with four different representations produced by simple bag-of-words, multifactor Course2vec, LSTM, and network-based embedding. We compared the performance of these pairings on five prediction tasks, course similarity, enrollment prediction, grade prediction, prerequisite prediction, and average enrollment prediction.

On the topic of the utility of syllabus data, which has not been evaluated before, we found that it showed benefit over catalog description data only in inferring prerequisite relationships (Recall of 0.5658 vs 0.5152), perhaps due to syllabi being the finer-grained source of content information about a course. In terms of course similarity signal, catalog description was markedly better than syllabus (Recall of 0.5372 vs 0.427) and our results indicate that catalog description, syllabus, and enrollment histories all bring some level of complementary information as the combination of all three performed better than any one or two combined. Enrollment data was used in the best scoring model in four of the five tasks, with only the best performing course similarity task model not utilizing enrollments. The nascent network-based approach performed well on all tasks, and was the top model in grade prediction and prerequisite prediction.

To conclude: (1) syllabus data is worth the effort to collect compared to catalog description for prerequisite prediction and (2) complements the catalog description and enrollment data on the course similarity task, (3) for prerequisite learning, supervised approaches based on embeddings perform much better than inferencing a pre-trained assessment model, (4) multifactor Course2vec often performs close to the more complex network-based approach on all tasks and (5) seeding the LSTM with course representations from the other models did not improve next-course prediction performance, while seeding with course grade representations from the student-course AMHEN model provided a small improvement in the grade prediction task.

8. LIMITATIONS AND FUTURE WORK

Our analyses were limited to data from a single large public institution in the US. Future work will need to evaluate multiple institutions of varying sizes, student demographics, and course taking policies in order to examine the generalizability of these approaches. In terms of models, we focused on simple text-based approaches and more complex neural models, both well established and nascent. Classical models of intermediary complexity were not evaluated.

We included tasks that have been common in EDM papers involving enrollment data; however, other institutional tasks exist that could be evaluated to produce an even more comprehensive analysis. These tasks include course preparation recommendation [21, 20], degree or course attrition prediction, and future course demand forecasting.

Syllabi in their original form could be evaluated, instead of in bag-of-words form, in order to investigate if the positionality of words in the syllabi offered any additional predictive utility. Lastly, learning management system clickstream data, as well as content information in addition to the syllabus, could be leveraged to enhance both content-based and collaborative-based course representations. This combination of different modalities and scales of data is an identified open challenge for the field [14].

9. ACKNOWLEDGEMENTS

We would like to thank Sandeep Jayaprakash and Oliver Heyer from UC Berkeley's Research Teaching, and Learning for supporting this research by facilitating the limited use of course syllabus data as pre-processed bag-of-words. We also thank the UC Berkeley Office of the Registrar for providing anonymized student enrollment data and access to course information APIs.

10. REFERENCES

[1] G. Angus, R. D. Martinez, M. L. Stevens, and A. Paepcke. Via: Illuminating academic pathways at scale. In *Proceedings of the Sixth ACM Conference on Learning@ Scale*, pages 1–10, 2019.

[2] C. Antunes. Acquiring background knowledge for intelligent tutoring systems. In *Proceedings of the 1st International Conference on Educational Data Mining*, 2008.

[3] M. G. Brown, R. M. DeMonbrun, and S. D. Teasley. Conceptualizing co-enrollment: Accounting for student experiences across the curriculum. In *Proceedings of the 8th International Conference on Learning Analytics and Knowledge*, pages 305–309, 2018.

[4] Y. Cen, X. Zou, J. Zhang, H. Yang, J. Zhou, and J. Tang. Representation learning for attributed multiplex heterogeneous network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1358–1368, 2019.

[5] S. Chang, W. Han, J. Tang, G.-J. Qi, C. C. Aggarwal, and T. S. Huang. Heterogeneous network embedding via deep architectures. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 119–128, 2015.

[6] W. Chen, A. S. Lan, D. Cao, C. Brinton, and M. Chiang. Behavioral analysis at scale: Learning course prerequisite structures from learner clickstreams. In *International Educational Data Mining Society*, 2018.

[7] M. Dillon. Introduction to modern information retrieval: G. salton and m. mcgill. mcgraw-hill, new york (1983). 448 pp., isbn 0-07-054484-0, 1983.

[8] M. Dong, R. Yu, and Z. A. Pardos. Design and deployment of a better course search tool: Inferring latent keywords from enrollment networks. In *European Conference on Technology Enhanced Learning*, pages 480–494. Springer, 2019.

[9] Y. Dong, N. V. Chawla, and A. Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 135–144, 2017.

[10] A. Elbadrawy and G. Karypis. Domain-aware grade prediction and top-n course recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 183–190, 2016.

[11] A. Elbadrawy and G. Karypis. Upm: Discovering course enrollment sequences associated with success. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*, pages 373–382, 2019.

[12] A. Esteban, A. Zafra, and C. Romero. A hybrid multi-criteria approach using a genetic algorithm for recommending courses to university students. In *International Educational Data Mining Society*, 2018.

[13] R. Farzan and P. Brusilovsky. Encouraging user participation in a course recommender system: An impact on user behavior. *Computers in Human Behavior*, 27(1):276–284, 2011.

[14] C. Fischer, Z. A. Pardos, R. S. Baker, J. J. Williams, P. Smyth, R. Yu, S. Slater, R. Baker, and M. Warschauer. Mining big data in education: Affordances and challenges. *Review of Research in Education*, 44(1):130–160, 2020.

[15] J. Gardner and C. Brooks. Coenrollment networks and their relationship to grades in undergraduate education. In *Proceedings of the 8th international conference on learning analytics and knowledge*, pages 295–304, 2018.

[16] Q. Hu and H. Rangwala. Academic performance estimation with attention-based graph convolutional networks. In *Proceedings of The 12th International Conference on Educational Data Mining*, volume 69, page 78. ERIC.

[17] L. Huang, C.-D. Wang, H.-Y. Chao, J.-H. Lai, and S. Y. Philip. A score prediction approach for optional course recommendation via cross-user-domain collaborative filtering. *IEEE Access*, 7:19550–19563, 2019.

[18] S. M. Jayaprakash, E. W. Moody, E. J. Lauría, J. R. Regan, and J. D. Baron. Early alert of academically at-risk students: An open source analytics initiative. *Journal of Learning Analytics*, 1(1):6–47, 2014.

[19] B. Jeon, E. Shafran, L. Breitfeller, J. Levin, and C. P. Rosé. Time-series insights into the process of passing or failing online university courses using neural-induced interpretable student states. In *Proceedings of the 12th International Conference on Educational Data Mining*, 2019.

[20] W. Jiang and Z. A. Pardos. Time slice imputation for personalized goal-based recommendation in higher education. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 506–510, 2019.

[21] W. Jiang, Z. A. Pardos, and Q. Wei. Goal-based course recommendation. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*, pages 36–45, 2019.

[22] P. Kaur, A. Polyzou, and G. Karypis. Causal inference in higher education: Building better curriculums. In *Proceedings of the Sixth ACM Conference on Learning@ Scale*, pages 1–4, 2019.

[23] Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang,

B. Zhou, and Y. Bengio. A structured self-attentive sentence embedding. *ICLR'17*, 2017.

[24] W. Liu, P.-Y. Chen, S. Yeung, T. Suzumura, and L. Chen. Principled multilayer network embedding. In *2017 IEEE International Conference on Data Mining Workshops*, pages 134–141. IEEE, 2017.

[25] Y. Luo and Z. A. Pardos. Diagnosing university student subject proficiency and predicting degree completion in vector space. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[26] H. Ma, X. Wang, J. Hou, and Y. Lu. Course recommendation based on semantic similarity analysis. In *2017 3rd IEEE International Conference on Control Science and Systems Engineering*, pages 638–641. IEEE, 2017.

[27] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[28] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[29] R. Morsomme and S. V. Alferez. Content-based course recommender system for liberal arts education. In *Proceedings of The 12th International Conference on Educational Data Mining*, volume 748, page 753. ERIC, 2019.

[30] S. Morsy and G. Karypis. Will this course increase or decrease your gpa? towards grade-aware course recommendation. *Journal of Educational Data Mining*, 11(2):20–46, 2019.

[31] Z. A. Pardos, H. Chau, and H. Zhao. Data-assistive course-to-course articulation using machine translation. In *Proceedings of the Sixth Conference on Learning@ Scale*, pages 1–10, 2019.

[32] Z. A. Pardos, Z. Fan, and W. Jiang. Connectionist recommendation in the wild: on the utility and scrutability of neural networks for personalized course guidance. *User Modeling and User-Adapted Interaction*, 29(2):487–525, 2019.

[33] Z. A. Pardos and W. Jiang. Designing for serendipity in a university course recommendation system. In *Proceedings of the Tenth International Conference on Learning Analytics & Knowledge*, pages 350–359. ACM, 2020.

[34] B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.

[35] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein. Deep knowledge tracing. In *Advances in neural information processing systems*, pages 505–513, 2015.

[36] A. Polyzou, N. Athanasios, and G. Karypis. Scholars walk: A markov chain framework for course recommendation. In *Proceedings of the 12th International Conference on Educational Data Mining*, pages 396–401, 2019.

[37] Z. Ren, X. Ning, A. Lan, and H. Rangwala. Grade prediction based on cumulative knowledge and co-taken courses. In *Proceedings of the 12th International Conference on Educational Data Mining*. ERIC, 2019.

[38] Z. Ren, X. Ning, and H. Rangwala. Grade prediction with temporal course-wise influence. In *International Educational Data Mining Society*, 2017.

[39] C. Romero, S. Ventura, P. G. Espejo, and C. Hervás. Data mining algorithms to classify students. In *Proceedings of the 1st International Conference on Educational Data Mining*, 2008.

[40] C. Yang, Z. Liu, D. Zhao, M. Sun, and E. Chang. Network representation learning with rich text information. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.