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-
The class at the end of the semester. The action of drop-

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

<table>
<thead>
<tr>
<th>Course grades</th>
<th>Grade prediction</th>
<th>Enrollment prediction</th>
<th>Prerequisite prediction</th>
<th>Course similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>[10, 15, 21, 38, 37, 16]</td>
<td>[10, 3, 32, 36]</td>
<td>[22, 11, 21, 16]</td>
<td>[17, 25, 12, 29]</td>
<td>[26, 31, 33, 12, 29]</td>
</tr>
<tr>
<td>Enrollment histories</td>
<td>[10, 15, 21, 38, 37, 16]</td>
<td>[10, 3, 32, 36, 1]</td>
<td>[22, 11, 21, 16]</td>
<td>[17, 25, 31, 33, 29]</td>
</tr>
<tr>
<td>Major declarations</td>
<td>[38, 21, 37]</td>
<td>[32, 36]</td>
<td>[21]</td>
<td>[25]</td>
</tr>
<tr>
<td>Catalog descriptions</td>
<td>[38, 21, 37]</td>
<td>[32, 36]</td>
<td>[21]</td>
<td>[25]</td>
</tr>
</tbody>
</table>

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 valuable 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 drop-

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 cam-
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>
<thead>
<tr>
<th>Method</th>
<th>Catalog descriptions</th>
<th>Course syllabus</th>
<th>Meta-information</th>
<th>Course enrollment histories</th>
<th>Course grades</th>
<th>Model type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bag-of-words</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>Static</td>
</tr>
<tr>
<td>Multi-C2v</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>Dynamic</td>
</tr>
<tr>
<td>LSTM</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>Dynamic</td>
</tr>
<tr>
<td>sc-AMHEN</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>Static</td>
</tr>
</tbody>
</table>

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.

\[
\sum_{i \in S} \sum_{c_i \in S} \sum_{w < j < w, j \neq 0} \log p(c_{i+j} | c_i, f_{i1}, f_{i2}, ..., f_{ih})
\]

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,

\[
p(c_{i+j} | c_i, f_{i1}, f_{i2}, ..., f_{ih}) = \frac{\exp(a_{i,j}^T v_{i+j})}{\sum_{h=1}^{h} \exp(a_{i,j}^T v_{i+h})}
\]

\[
a_i = v_i + \sum_{j=1}^{h} W_{nj \times s} f_{ij}
\]

where \( a_c \) is the vector sum of input course vector representation \( v_c \) and all the features vector representations of course \( c, f_{ij} \) is the multi-hot input of the \( j \)-th feature of course \( i \), and \( W_{nj \times v} \) is the weight matrix for feature \( j \). So by multiplying \( W_{nj \times v} \) and \( f_{ij} \), 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. \( v_i \) 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 \( W_f, W_k, W_r, \) and \( W_c \) learned by the LSTM transferred the
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.

In the student-course attributed multiplex heterogeneous network we described above, \( V = (C, S) \), where each node \( c \in C \) represents a course in the course set \( C \) and each node \( s \in S \) represents a student in the student set \( S \). \( R \) 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 \( G = (C, S, E, A) \), the problem of Student-Course AMHEN embedding is to give a unified low-dimensional space representation of each student node \( s \in S \) and each course node \( c \in C \) on every grade type \( r \). The goal is to find a function \( g : S \to \mathbb{R}^d \) and a function \( f_r : C \to \mathbb{R}^d \) 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 \( b_c \), grade embedding \( g \), and attribute embedding \( u \), and split the overall student embedding into two parts: base embedding \( b_s \), and individual embedding \( p \).

The base embedding of course node \( c_i \), i.e., \( b_{c_i} \), is shared between different grade types. We define \( b_{c_i} \) as a parameterized function of \( c_i \)’s attributes \( x_i \in \mathbb{R}^d \) as:

\[
b_{c_i} = h(x_i) \tag{4}
\]

where \( h \) is a transformation function, such as a multi-layer perceptron. The attribute embedding of course node \( c_i \), i.e., \( u_i \), is defined as:

\[
u_i = D^T x_i \tag{5}
\]

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 \( g_{cr}^{(k)} \in \mathbb{R}^d \), \( 1 \leq k \leq K \) 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 \( c_i \).

\[
g_{cr}^{(k)} = \text{mean}\{p_j^{(k-1)} \mid \forall p_j \in N_i \} \tag{6}
\]

Similarly, the \( k \)-th level of individual embedding \( p_i^{(k)} \in \mathbb{R}^d \), \( 1 \leq k \leq K \) 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.

\[
p_i^{(k)} = \text{mean}\{g_{cr}^{(k)} \mid \forall c_j \in N_{si} \} \tag{7}
\]

\(^1\)By level we mean iteration, i.e., the embedding is updated after each parameters update process.

---

**Definition 1.** (Attributed Multiplex Heterogeneous Network): An attributed multiplex heterogeneous network is a network \( G = (V, E, A) \), \( E = \bigcup_{r \in R} E_r \), where \( E_r \) consists of all edges with edge type \( r \in R \), and \( |R| > 1 \). We separate the network for every edge type \( r \in R \) as \( G_r = (V, E_r, A) \). Each node \( v_i \in V \) is associated with some types of feature vectors. \( A = \{ x_i \mid v_i \in V \} \) is the set of node features for all nodes, where \( x_i \) is the associated node feature of node \( v_i \).
We denote the k-th level grade embedding $g_{ir}^{(k)}$ as grade embedding $g_{ir}$, and concatenate all the grade embeddings for course node $c_i$ as $G_i \in \mathbb{R}^{d \times m}$, where $d$ is the dimension of grade embeddings and $m$ is the number of grade types.

$$G_i = (g_{i1}, g_{i2}, ..., g_{im})$$  

(8)

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

$$a_{ir} = \text{softmax}(u_r^T \tanh(W_r G_i))^T$$  

(9)

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

$$c_{ir} = \alpha_r h(x_i) + M_r^T G_i a_{ir} + \beta_r D_r^T x_i$$  

(10)

where $M_r \in \mathbb{R}^{d \times n}$ and $D_r \in \mathbb{R}^{n \times n}$ are trainable transformation matrix. $\alpha_r$ and $\beta_r$ are two coefficients adjusting the weights of the three embeddings of courses, which can also be trainable.

The overall embedding of student node $s_j$ is:

$$s_j = \alpha_s b_s + N^T p_i$$  

(11)

where $\alpha_s$ is a trainable coefficient adjusting the weights of the two embeddings of students, and $N \in \mathbb{R}^{d \times n}$ 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(type) type, then given a view (grade type) $r$ of the network, i.e., $G_r = (C, S, \mathcal{E}_r, A)$, 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:

$$P(s_j|c_i, r) = \frac{\exp(c_{ir}' s_j^*)}{\sum_{k \in S} \exp(c_{ir}' s_k^*)}$$  

(12)

where $s_{k}^*$ is the context embedding of student node $s_k$. For a training pair $(s_j, c_i)$ with grade type $r$, our objective is to maximize the probability:

$$P(c_j|s_i, r) = \frac{\exp(s_{i}^T c_{jr}^*)}{\sum_{k \in C} \exp(s_{i}^T c_{kjr}^*)}$$  

(13)

where $c_{jr}^*$ 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

$$\text{loss}(c_i, s_j, r) = -\log \sigma(c_{ir}' s_j^*) - \sum_{l=1}^{L} \mathbb{E}_{s_k \sim P(s_k)}[\log \sigma(-c_{ir}' s_k^*)]$$  

(14)
and the objective function $-\log P(c_j|s_i, r)$ for node pair $(s_i, c_j)$ as:

$$
\text{loss}(s_i, c_j, r) = -\log \sigma(s^T_i c_j^r) - \sum_{t=1}^{L} \mathbb{E}_{s_k \sim P(s_k)}[\log \sigma(-s^T_k c_{k+1}^r)]
$$

Here we define $P(s_k) = \frac{f(s_k)^{3/4}}{\sum_{i=1}^{C} f(s_i)^{3/4}}$ 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 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.

$$
c_i = (c_{i1}^T, c_{i2}^T, \ldots, c_{im}^T)^T
$$

5. TASKS

In this section, we describe five downstream 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, the dimensions of the model input in each time slice is reset to 0 if rank $t > 10$. 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 $\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{Q} \frac{1}{\text{rank}_i}$, where $\text{rank}_i$ 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$_i > 10$.

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 $\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{Q} \frac{1}{\text{rank}_i}$, 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$_i > 10$.

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 $g_i$ represents the grades of courses taken in semester $i$ and $c_i$ represents the courses taken in semester $i$. $c_{i+1}$ is concatenated with $g_i$ 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.

$$
g(s_i, c_j) = \arg \max_{c} \cos(s_i, c_j)
$$

Figure 3: Illustration of the LSTM-based next-course 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 inferring 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 inferred in an unsupervised manner (i.e., not using any prerequisite ground truth), to predict course prerequisites.

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

<table>
<thead>
<tr>
<th>Representation created by</th>
<th>Course similarity prediction</th>
<th>Enrollment prediction</th>
<th>Grade prediction</th>
<th>Prerequisite prediction</th>
<th>Avg. enroll-predict</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>Course similarity Mean/Median Rank</td>
<td>Recall@10</td>
<td>Recall@10</td>
<td>MRR@10</td>
<td>Accuracy</td>
</tr>
<tr>
<td>bag-of-words</td>
<td>catalog</td>
<td>602/6</td>
<td>0.5370</td>
<td>0.3154</td>
<td>0.5216</td>
</tr>
<tr>
<td>bag-of-words</td>
<td>syllabus</td>
<td>329/19</td>
<td>0.4270</td>
<td>0.3744</td>
<td>0.5103</td>
</tr>
<tr>
<td>multi-c2v</td>
<td>enrollments, course meta-information</td>
<td>224/15</td>
<td>0.4485</td>
<td>0.3791</td>
<td>0.5576</td>
</tr>
<tr>
<td>LSTM (multi-hot)</td>
<td>enrollments</td>
<td>584/58</td>
<td>0.2924</td>
<td>0.3967</td>
<td>0.5885</td>
</tr>
<tr>
<td>sc-AMHEN</td>
<td>enrollments, grades, catalog</td>
<td>288/11</td>
<td>0.4767</td>
<td>0.3882</td>
<td>0.5625</td>
</tr>
</tbody>
</table>

Table 4: Course similarity validation of all the course representations

<table>
<thead>
<tr>
<th>Model</th>
<th>Mean/Median Rank</th>
<th>Recall @10</th>
</tr>
</thead>
<tbody>
<tr>
<td>catalog</td>
<td>602/6</td>
<td>0.5372</td>
</tr>
<tr>
<td>syllabus</td>
<td>329/19</td>
<td>0.4270</td>
</tr>
<tr>
<td>course2vec (c2v)</td>
<td>224/21</td>
<td>0.3839</td>
</tr>
<tr>
<td>multi-c2v (mc2v)</td>
<td>224/15</td>
<td>0.4485</td>
</tr>
<tr>
<td>catalog+mc2v</td>
<td>132/3</td>
<td>0.6435</td>
</tr>
<tr>
<td>syllabus+mc2v</td>
<td>109/6</td>
<td>0.5798</td>
</tr>
<tr>
<td>catalog+syllabus+mc2v</td>
<td>79/3</td>
<td>0.6705</td>
</tr>
<tr>
<td>catalog+syllabus+mc2v</td>
<td>(PCA dim: 300)</td>
<td>177/3</td>
</tr>
<tr>
<td>LSTM</td>
<td>584/58</td>
<td>0.2924</td>
</tr>
<tr>
<td>sc-AMHEN(u)</td>
<td>288/11</td>
<td>0.4767</td>
</tr>
<tr>
<td>sc-AMHEN(c)</td>
<td>330/27</td>
<td>0.3603</td>
</tr>
</tbody>
</table>

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.

6.2 Grade Prediction

The accuracy of the grade predictions generated by the pure student-course AMHEN model (sc-AMHEN(s,c)), the LSTM model with multi-hot as input (LSTM(multi-hot)), and the LSTM model with course embeddings with different grade types (LSTM(u,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 information (semester order) of the grades of enrolled courses. The grade prediction results show that the graph model, though static, could map the knowledge
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 superiority 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.

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 inferring 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


