

Designing for Serendipity in a University Course Recommendation System

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

Weijie Jiang
Tsinghua University
Beijing, China
jiangwj.14@sem.tsinghua.edu.cn

ABSTRACT

Collaborative filtering based algorithms, including Recurrent Neural Networks (RNN), tend towards predicting a perpetuation of past observed behavior. In a recommendation context, this can lead to an overly narrow set of suggestions lacking in serendipity and inadvertently placing the user in what is known as a "filter bubble." In this paper, we grapple with the issue of the filter bubble in the context of a course recommendation system in production at a public university. Our approach is to present course results that are novel or unexpected to the student but still relevant to their interests. We build one set of models based on course catalog descriptions (BOW) and another set informed by enrollment histories (course2vec). We compare the performance of these models on off-line validation sets and against the system's existing RNN-based recommendation engine in an online user study of undergraduates ($N = 70$) who rated their course recommendations along six characteristics related to serendipity. Results of the user study show a dramatic lack of novelty in RNN recommendations and depict the characteristic trade-offs that make serendipity difficult to achieve. While the machine learned course2vec models performed best on off-line validation tasks, it was the simple bag-of-words based recommendations that students rated as more serendipitous. We discuss the role of the kind of information presented by the system in a student's decision to accept a recommendation from either algorithm.

CCS CONCEPTS

• Applied computing → Education; • Information systems → Recommender systems.

KEYWORDS

Higher education, course guidance, filter bubble, neural networks

ACM Reference Format:

Zachary A. Pardos and Weijie Jiang. 2020. Designing for Serendipity in a University Course Recommendation System. In *Proceedings of the 10th International Conference on Learning Analytics and Knowledge (LAK '20)*, March 23-27, 2020, Frankfurt, Germany. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3375462.3375524>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

LAK '20, March 23-27, 2020, Frankfurt, Germany

© 2020 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-7712-6/20/03.
<https://doi.org/10.1145/3375462.3375524>

1 INTRODUCTION

Among the institutional values of a liberal arts university is to expose students to a plurality of perspectives expressed in courses across its various physical and intellectual schools of thought. Collaborative filtering based sequence prediction methods, in this environment, can provide personalized course recommendations based on temporal models of normative behavior [30] but are not well suited for surfacing courses a student may find interesting but which have been relatively unexplored by those with similar course selections to them in the past. Therefore, a more diversity oriented model can serve as an appropriate compliment to recommendations made from collaborative based methods. This problem of training on the past without necessarily repeating it is an open problem in many collaborative filtering based recommendation contexts, particularly social networks, where, in the degenerate cases, users can get caught in "filter bubbles," or model-based user stereotypes, leading to a narrowing of item recommendation variety [20, 28, 40]. In many educational contexts this might manifest as a type of swim laning of a student into a particular track based on a machine learned stereotype formed. **Learning analytics decision support systems, whether teacher, administrator, or student facing, should be cognisant of this phenomena and design accordingly** [7].

To counteract the filter bubble, we introduce a course2vec variant into a production recommender system at a public university designed to surface serendipitous course suggestions. Course2vec applies a skip-gram to course enrollment histories, instead of natural language, in order to learn course representations. We use the definition of serendipity as user perceived unexpectedness of result combined with successfulness [14, 35], which we define as a course recommendation a student expresses interest in taking. At many universities, conceptually similar courses exist across departments but use widely differing disciplinary vernacular in their catalog descriptions, making them difficult for learners to search for and to realize their commonality. We propose that by tuning a vector representation of courses learned from nine years of enrollment sequences, we can capture enough implicit semantics of the courses to more abstractly, and accurately construe similarity. To encourage the embedding to learn features that may generalize across departments, our skip-gram variant simultaneously learns department (and instructor) embeddings. While more advanced attention-based text generation architectures exist [38], we demonstrate that properties of the linear vector space produced by "shallow" networks are of utility to this recommendation task. Our recommendations are made with only a single explicit course preference given by the user, as opposed to the entire course selection history needed by session-based Recurrent Neural Network approaches [15]. Single example, also known as "one-shot," generalization is common in the vision

community, which has pioneered approaches to extrapolating a category from a single labeled example [13, 39]. Other related work applying skip-grams to non-linguistic data include node embeddings learned from sequences of random walks of graphs [34] and product embeddings learned from ecommerce clickstream [8]. Our work, methodologically, adds rigor to this approach by **tuning the model against validation sets created from institutional knowledge and curated by the university**.

We conduct a user study ($N = 70$) of undergraduates at the University to evaluate their personalized course recommendations made by our models designed for serendipity and by the RNN-based engine, which previously drove recommendations in the system. The findings underscore the tension between unexpectedness and successfulness and show the deficiency of RNNs for producing novel course recommendations. While the *course2vec*-based model scored substantially above bag-of-words (BOW) model in accuracy on off-line validation sets, BOW scored slightly higher in the main objective of user perceived serendipity. A potential reason for this discrepancy is the nature of information presented to students in the recommender system interface. Catalog descriptions of recommended courses were shown to students, which served as the only source of information they could consult in deciding if they wanted to take the course. Conveying the latent features of a recommended course picked up by the machine learned model may be necessary to realize its benefit.

2 RELATED WORK

In Natural Language Processing, a classical representation of words is as a vector of the contexts they appear in. Such vector representations are called explicit, as each dimension directly corresponds to a particular context [22]. These explicit vector-space representations have been extensively studied in the NLP literature [5, 37], and are known to exhibit a large amount of attributional similarity [21, 23]. Although Baroni et al. [4] show that the neural embeddings obtain a substantial improvement against explicit representations on a wide range of lexical semantics tasks, Levy and Goldberg [22] argue that under certain conditions, traditional word similarities induced by explicit representations can perform just as well on lexical relationship validation sets. Their debates encourage us to utilize course descriptions to generate explicit bag-of-words representations for courses and compare them to our neural embedding models.

Nguyen et al. [28] measured the "filter bubble" effect in terms of content diversity at the individual level and found that collaborative filtering-based recommender systems expose users to a slightly narrowing set of items over time. McNee et al. [25] also proposed that the recommender community should move beyond conventional accuracy metrics and their associated experimental methodologies. To counter the "filter bubble", Zhang et al. [40] used a collection of novel LDA-based algorithms inspired by principles of "serendipitous discovery" and injected serendipity, novelty, and diversity to music recommendations whilst limiting the impact on accuracy. Adamopoulos and Tuzhilin [3] proposed a new concept of unexpectedness as recommending to users those items that depart from what they would expect from the system. Different serendipitous metrics that measure the uncertainty and relevance of user's attitude towards items in order to mitigate the over-specialization

problem with surprising suggestions are combined with traditional collaborative filtering recommendation [16] and content-based recommendation [1]. Kawamae et al. [19] presented the Personal Innovator Degree (PID) which focused on the dynamics and precedence of user preference to recommend items that match the latest preference of the target user to achieve serendipity.

Recommender systems in higher education contexts have recently focused on prediction of which courses a student will take [30, 32] or the grade they will receive if enrolled [17, 18, 33]. At Stanford, a system called "CARTA" allows students to see grade distributions, course evaluations, and the most common courses taken before a course of interest [6]. At UC Berkeley, the recommender system being modified in this study serves students next-semester course considerations based on their personal course enrollment history [30]. Earlier systems included a focus on requirement satisfaction [29] and career-based relevancy recommendation [12]. No system has yet focused on serendipitous or novel course discovery, important in US higher education due to the substantial elective course choice afforded to students [2, 36].

3 MODELS AND METHODOLOGY

This section introduces three competing models used to generate our representations. The first model uses *course2vec* [30] to learn course representations from enrollment sequences. Our second model is a variant on *course2vec*, which learns representations of explicitly defined features of a course (e.g., instructor and department) in addition to the course representation. The intuition behind this approach is that the course representation could have, conflated in it, the influence of the multiple-instructors that have taught the course over time. We contend that this "deconflation" may increase the fidelity of the course representation and serve as a more accurate representation of the topical essence of the course. The last representation model is a standard bag-of-words vector, constructed for each course strictly from its catalog description. Finally, we explore concatenating a course's *course2vec* and bag-of-words representation vector.

3.1 Course2vec

The *course2vec* model involves learning 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 linguistic domain. For each student s , their chronological course enrollment sequence is produced by first sorting by semester then randomly serializing within-semester course order. Then, each course enrollment sequence is trained on like a sentence in the Skip-gram model.

In language models, two word vectors will be cosine similar if they share similar sentence contexts. Likewise, in the university domain, courses that share similar co-enrollments, and similar previous and next semester enrollments, will likely be close to one another in the vector space. *Course2vec* learns course representations using a skip-gram model [27] by maximizing the objective function over all the students' course enrollment sequences.

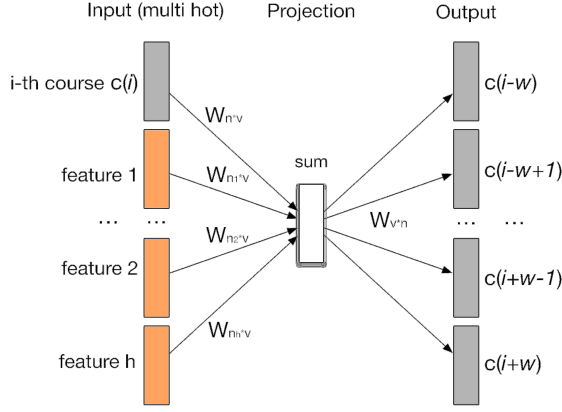


Figure 1: multi-factor course2vec model

3.2 Multi-factor Course2vec

The training objective of the skip-gram model is to find word representations that are useful for predicting the surrounding words in a sentence or a document. Each word in the corpus is used as an input to a log-linear classifier with continuous projection layer, to predict words within a certain range before and after the current word. Therefore, the skip-gram model can be also deemed as a classifier with input as a target course and output as a context course. In this section, we consider adding more features of courses to the input to enhance the classifier and its representations, as shown in Figure 1. Each course is taught by one or several instructors over the years and is associated with an academic department. The multi-factor course2vec model learns both course and feature representations by maximizing the objective function over all the students' enrollment sequences and the features of courses, defined as follows.

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

Probability $p(c_{i+j} | c_i, f_{i1}, f_{i2}, \dots, f_{ih})$ of observing a neighboring course c_{i+j} given the current course c_i and its features $f_{i1}, f_{i2}, \dots, f_{ih}$ can also be defined via the softmax function,

$$p(c_{i+j} | c_i) = \frac{\exp(a_i^T \mathbf{v}'_{i+j})}{\sum_{k=1}^n \exp(a_i^T \mathbf{v}'_k)} \quad (2)$$

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

where \mathbf{a}_i is the vector sum of input course vector representation \mathbf{v}_i and all the features vector representations of that course, \mathbf{f}_{ij} is the multi-hot input of the j -th feature of course i , and $\mathbf{W}_{n_j \times v}$ is the weight matrix for feature j . So by multiplying $\mathbf{W}_{n_j \times v}$ and \mathbf{f}_{ij} , it gets the sum of feature vector representations of the i -th course. Here we keep the vector dimensions the same for both courses and features so that they are learned and mapped to the same space.

3.3 Bag-of-Words

A simple but indelible approach to item representation has been to create a vector, the length of the number of unique words across all

items, with a non-zero value if the word in the vocabulary appears in it. Only unigram words are used to create this unordered vector list of words used to represent the document [9].

The basic methodology based on bag-of words proposed by IR researchers for text corpora - a methodology successfully deployed in modern Internet search engines - reduces each document in the corpus to a vector of real numbers, each of which represents a term weight. The term weight might be:

- a term frequency value indicating how many times the term occurred in the document.
- a binary value with 1 indicating that the term occurred in the document, and 0 indicating that it did not.
- tf-idf scheme [10], the product of term frequency and inverse document frequency, which increases proportionally to the number of times a word appears in the document and is offset by the frequency of the word in the corpus and helps to adjust for the fact that some words appear more frequently in general.

We evaluate all three variants in our quantitative validation testing.

3.4 Surfacing Serendipitous Recommendations from Course Representations

We surface recommendations intended to be interesting but unexpected by finding an objective course c_j which is most similar to a student's favorite course c_i , diversifying the results by allowing only one result per department d_j :

$$c_j^* = \arg \max_{c, d(c)=d_j} \cos(c, c_i) \quad (4)$$

where $d(c)$ means the the department of course c . Then all the counterpart courses c_j^* in all the other departments will be ranked according to $\cos(c_j^*, c_i)$, where $j = 1, 2, \dots, k$. We can apply both neural representations and bag-of-words representations of courses in this method to generate the most similar courses in each department.

4 EXPERIMENTAL ENVIRONMENTS

4.1 Off-line Dataset

We used a dataset containing anonymized student course enrollments at UC Berkeley from Fall 2008 through Fall 2017. The dataset consists of per-semester course enrollment records for 164,196 students (both undergraduates and graduates) with a total of 4.8 million enrollments. A course enrollment record means that the student was still enrolled in the course at the end of the semester. Students at this university, during this period, were allowed to drop courses up until close to the end of the semester without penalty. The median course load during students' active semesters was four. There were 9,478 unique lecture courses from 214 departments¹ hosted in 17 different Divisions of six different Colleges. Course meta-information contains course number, department name, total enrollment and max capacity. In this paper, we only consider lecture courses with at least 20 enrollments total over the 9-year period, leaving 7,487 courses. Although courses can be categorized

¹ At UC Berkeley, the smallest academic unit is called a "subject." For the purpose of communicability, we instead refer to subjects as departments.

as undergraduate courses and graduate courses, undergraduates are permitted to enroll in many graduate courses no matter their status.

Enrollment data were sourced from the campus enterprise data warehouse with course descriptions sourced from the official campus course management system API. We pre-processed the course description data in the following steps: (1) removing generic, often-seen sentences across descriptions (2) removing stop words (3) removing punctuation (4) word lemmatization and stemming, and finally tokenizing the bag-of-words in each course description. We then compiled the term frequency vector, binary value vector, and tf-idf vector for each course.

4.1.1 Semantic Validation Sets. In order to quantitatively evaluate how accurate the vector models are, a source of ground truth on the relationships between courses needed to be brought to bear to see the degree to which the vector representations encoded this information. We used two such sources of ground truth to serve as validation sets, one providing information on similarity, the other on a variety of semantic relationships.

- *Equivalency validation set:* A 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. For example, an honors and non-honors version of a 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.
- *Analogy validation set:* The standard method for validating learned word vectors has been to use analogy to test the degree to which the embedding structure contains semantic and syntactic relationships constructed from prior knowledge. In the domain of university courses, we use course relationship pairs constructed from prior work using first-hand knowledge of the courses [31]. The 77 relationship pairs were in five categories; online, honors, mathematical rigor, 2-department topics, and 3-department topics. An example of an "online" course pair would be Engineering 7 and its online counterpart, Engineering W7 or Education 161 and W161. An analogy involving these two pairs could be calculated as: Engineering 7W - Engineering 7 + Education 161 \approx Education W161.

4.2 Online Environment (System Overview)

The production recommender system at UC Berkeley uses a student data pipeline with the enterprise data warehouse to keep up-to-date enrollment histories. Upon CAS login, these histories are associated with the student and passed through an RNN model, which cross-references the output recommendations with the courses offered in the target semester. Class availability information is retrieved during the previous semester from a campus API once the registrar has released the schedule. The system is written with an AngularJS front-end and python back-end service which loads the machine learned models written in pyTorch. These models are version controlled on github and refreshed three times per semester after student enrollment status refreshes from the pipeline. The

system receives traffic of around 15% (4,500) of the undergraduate student body, partly from the Registrar's website.

5 VECTOR MODEL REFINEMENT EXPERIMENTS

In this section, we first introduce our experiment parameters and the ways we validated the representations quantitatively. Then, we describe the various ways in which we refined the models and the results of these refinement.

5.1 Model Evaluations

We trained the models described in Section 3 on the students enrollment records dataset. Specifically, we added the instructor(s) who taught the course and the course department as two features of courses in the multi-factor course2vec model. We trained the models and tuned the parameters via our Pytorch implementation². Given that the size of the course vocabulary (7.5K) is smaller than that of a language vocabulary (692K) [27], we explored smaller vector sizes of 100, 200 and 300 but find 300 performs best for course2vec, which is the same as that of word2vec [26]. We kept the vector size 300 for the multi-factor course2vec models. We applied two kinds of established extensions that improve both the quality of the vectors and the training speed in word2vec; subsampling of the frequent words and an approximation to softmax called negative sampling [27]. The best subsampling factor for our model is 10^{-3} , compared to 10^{-5} for word2vec, while negative sampling performs worse than softmax. The reason for a higher subsampling threshold likely lies in that there are not as many courses that appear frequently in students' enrollment sequences as the stop words that commonly occurred in the corpus, thus only courses with frequencies greater than 10^{-3} , which is higher than 10^{-5} , should be subsampled. It is not surprising that softmax might perform better as it applies back-propagation to all word representations at every observation instead of sampling the negative words (not in context) to update.

To evaluate course vectors 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 described the performance of each model on all validation pairs in terms of *mean rank*, *median rank* and *recall@10*.

For evaluation of the course analogy validation set, we followed the analogy paradigm of: $course2 - course1 + course3 \approx course4$. Courses were ranked by their cosine similarity to $course2 - course1 + course3$. An analogy completion is considered accurate (a hit) if the first ranked course is the expected *course4* (excluding the other three from the list). We calculated the average accuracy (recall@1) and the recall@10 over all the analogies in this validation set.

5.2 Course2vec vs. Multi-factor Course2vec

We compared the pure course2vec model with the course representations from the multi-factor course2vec model using instructor, department, and both as factors. Results of evaluation on the equivalency validation are shown in Table 1 with analogy validation

²<https://github.com/CAHLR/Serendipitous-Course-Recommendation>

Table 1: (Offline) Equivalency validation results

Model	Mean / Median Rank	Recall @10
course2vec	244 / 21	0.3839
ins-course2vec	302 / 16	0.4406
dept-course2vec	261 / 17	0.4005
ins-dept-course2vec	224 / 15	0.4485
tf	627 / 8	0.5164
binary	652 / 9	0.5012
tf-idf	602 / 6	0.5372
tf+insdept-course2vec	168 / 6	0.5691
tf+insdept-course2vec (norm)	132 / 3	0.6371
bin.+insdept-course2vec	178 / 7	0.5404
bin.+insdept-course2vec (norm)	129 / 3	0.6251
tfidf+insdept-course2vec	213 / 14	0.4428
tfidf+insdept-course2vec (norm)	132 / 3	0.6435

Table 2: (Offline) Analogy validation results

Model	Accuracy	Recall@10
course2vec	0.5011	0.7685
ins-course2vec	0.5138	0.7853
dept-course2vec	0.3581	0.8284
ins-dept-course2vec	0.4961	0.8557
tf	0.3037	0.537
binary	0.3159	0.581
tf-idf	0.3227	0.542
tf+insdept-course2vec	0.5066	0.8438
tf+insdept-course2vec (norm)	0.448	0.6872
bin.+insdept-course2vec	0.5193	0.8788
bin.+insdept-course2vec (norm)	0.4603	0.7449
tfidf+insdept-course2vec	0.5138	0.8584
tfidf+insdept-course2vec (norm)	0.4503	0.7059

results shown in Table 2, where bold values represent the best in their sections. The multi-factor model outperformed the pure course2vec model in terms of recall@10 in both validation sets, with the combined instructor and department factor model performing the best. The same result held for the metrics of mean and median rank in equivalency, but multi-factor models were not always the best in terms of analogy recall@1 (Accuracy). The best recall achieved in analogies was by the instructor and department multi-factor model, successfully completing 85.57% of course analogies when considering the top 10 candidates of each analogy completion.

5.3 Bag-of-words vs. Multi-factor Course2vec

Among the three bag-of-words models, tf-idf performed the best in all equivalency set metrics, as seen in Table 1. The median rank (best=6) and recall@10 (best=0.5372) for the bag-of-words models were also substantially better than the best course2vec models, which had a best median rank of 15 with best recall@10 of 0.4485 for the multi-factor instructor and department model. All course2vec models; however, showed better mean rank performance (best=224) compared with bag-of-words (best=602). This suggests that there are many outliers where literal semantic similarity (bag-of-words)

is very poor at identifying equivalent pairs, whereas course2vec has much fewer near worst-case examples. This result is consistent with prior work comparing pure course2vec models to binary bag-of-words [30].

When considering performance on the analogy validation (Table 2), the roles are reversed, with all course2vec models performing better than the bag-of-words models in both accuracy and recall@10. The difference in recall of bag-of-words compared to course2vec when it comes to analogies is substantial (0.581 vs 0.8557), a considerably larger difference than between bag-of-words and course2vec on equivalencies (0.5372 vs 0.4485). Again, the multi-factor instructor and department model and tf-idf were the best models in their respective class. These analyses establish that bag-of-words models are moderately superior in capturing course similarity, but are highly inferior to enrollment-based course2vec models in the more complex task of analogy completion.

5.4 Combining Bag-of-words and Course2vec Representations

In light of the strong analogy performance of course2vec and strong equivalency performance bag-of-words in the previous section, we concatenated the multi-factor course2vec representations with bag-of-words representations. To address the different magnitudes in the vectors between the two concatenated representations, we create a normalized version of each vector set for comparison to non-normalized sets.

We found that the normalized concatenation of tf-idf with multi-factor course2vec performed substantially better on the equivalency test than the previous best model in terms of recall@10 (0.6435 vs. 0.5372) as seen in Table 1. While the median rank of the concatenated model only improved from 6 to 3, the mean rank improved dramatically (from 602 to 132), and is the best of all models tested in terms of mean rank. Non-normalized vectors did not show improvements over bag-of-words alone in median rank and recall@10. Improvements in the analogy test were more mild, with a recall@10 of 0.8788 for the best concatenated model, combining binary bag-of-words with multi-factor course2vec, compared with 0.8557 for the best course2vec only model. Normalization in the case of analogies hurt all model performance, the opposite of what was observed in the equivalency test. This suggests that normalization improves local similarity but may act to degrade the more global structure of the vector space.

6 USER STUDY

A user study was conducted to evaluate the quality of recommendations drawn from our different course representations. **Users rated each course from each recommendation algorithm along five dimensions of quality.** Students were asked to rate course recommendations in terms of their (1) unexpectedness (2) successfulness / interest in taking the course (3) novelty (4) diversity of the results (5) and identifiable commonality among the results. In Shani and Gunawardana [35], authors defined serendipity as the combination of "unexpectedness" and "success." In the case of a song recommender, for example, success would be defined as the user listening to the recommendation. In our case, we use a student's expression of interest in taking the course as a proxy for success. The mean

of their unexpectedness and successfulness rating will comprise our measure of serendipity. We evaluated three of our developed models, all of which displayed 10 results, only showing one course per department in order to increase diversity (and unexpectedness). The models were (1) the best *BOW* model (tf-idf), (2) the best *Analogy* validation model (binary BOW + multi-factor course2vec normalized), (3) and the best *Equivalency* validation model (tf-idf + multi-factor course2vec non-normalized). To measure the impact our department diversification filter would have on serendipity, we added a version of the best *Equivalency* model that did not impose this filter, allowing multiple courses to be displayed from the same department if they were the most cosine similar to the user's specified favorite course. Our fifth comparison recommendation algorithm was the system's existing collaborative-filtering based Recurrent Neural Network (RNN) that recommends courses based on a prediction of what the student is likely to take next given their personal course history and what other students with a similar history have taken in the past [30]. All five algorithms were integrated into a real-world recommender system for the purpose of this study and evaluated by 70 undergraduates at the University.

6.1 Study Design

Undergraduates were recruited from popular University associated Facebook groups and asked to sign-up for a one hour evaluation session. Since they would need to specify a favorite course they had taken, we restricted participants to sophomore level students and above. The study was run at the beginning of the Fall semester, while courses could still be added and dropped and some students were still shopping for courses. We used a within-subjects design whereby each participant rated ten course recommendations made by each of the five algorithms. The order in which students saw recommendations from the four serendipitous algorithms introduced in this study was randomized. The fifth algorithm, the RNN, was always last. This algorithm's recommendations, personalized to the student's enrollment history, had very little overlap with serendipity algorithms' recommendations, which were personalized only to the input of a favorite course. Because of the considerable number of ratings expected ($[3 \times 10 + 2] \times 5 = 160$) and the importance for students to carefully consider each recommended course, in-person sessions were decided on over asynchronous remote sessions in order to better encourage on-task behavior throughout the session. Student evaluators were compensated with a \$40 gift card to attend one of four sessions offered across three days with a maximum occupancy of 25 in each session. A total of 70³ students participated. Their majors skewed heavily STEM, with 25 of the 70 majoring in computer science and/or electrical engineering. Economics was the second most common major, with five students represented.

We began the session by introducing the evaluation motivation as a means for students to help inform the choice of algorithm that we will use for a future campus-wide deployment of a course exploration tool. Students started the evaluation by visiting a survey URL that asked them to specify a favorite course they had

³Due to an authentication bug during the fourth session, twenty participating students were not able to access the collaborative recommendations of the fifth algorithm. RNN results in the subsequent section are therefore based on the 50 students from the first three sessions. When post hoc significance tests are conducted between RNN and the ratings of other algorithms, the tests are between ratings among these 50 students.

Algorithm 1
Please follow [THIS](#) search link and enter in your selected favorite course. Ten courses will be shown intended to have some relation to the course you entered.
(If any other search results are open in another browser window or tab, please close them)

* 3. Based on the results from the link above, rate your agreement with the statement in the columns of the table below for each of the 10 course results. If you are not familiar with the course, be sure to read the included course description in full to help inform your ratings.

	This course was unexpected	I am interested in taking this course	I did not know about this course before
Course #1	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
⋮	⋮	⋮	⋮
Course #10	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

* 4. Rate your agreement with the following statement

	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree
Overall, the course results were diverse	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
The course results shared something in common with my favorite course	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

5. If you identified something in common with your favorite course, please explain it here.

Figure 2: User survey page

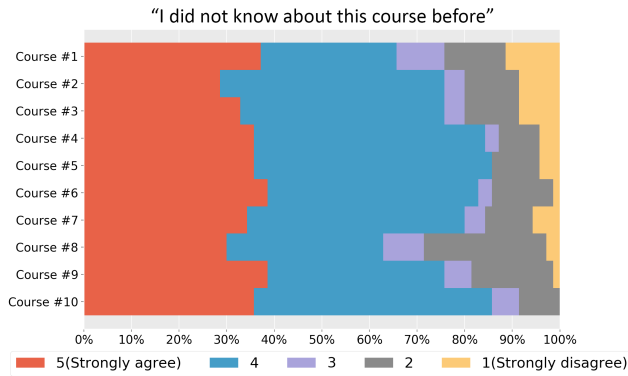
taken at the University. This favorite course was used by the first four algorithms to produce 10 course recommendations each. Each recommended course's department, course number, title, and full catalog description were displayed to the student in the interface. There was a survey page (Figure 2) for each algorithm in which students were asked to read the recommended course descriptions carefully and then rate each of the ten courses individually on a five point Likert scale agreement with the following statements: (1) This course was unexpected (2) I am interested in taking this course (3) I did not know about this course before. These ratings respectively measured unexpectedness, successfulness, and novelty. After rating the individual courses, students were asked to rate their agreement with the following statements pertaining to the 10 results as a whole: (1) Overall, the course results were diverse (2) The course results shared something in common with my favorite course. These ratings measured dimensions of diversity and commonality. Lastly, students were asked to provide an optional follow-up open text response to the question, "If you identified something in common with your favorite course, please explain it here." On the last page of the survey, students were asked to specify their major, year, and to give optional open response feedback on their experience. Graduate courses were not included in the recommendations and the recommendations were not limited to courses available in the current semester.

6.2 Results

Results of average student ratings of the five algorithms across the six measurement categories are shown in Table 3. The diversity based algorithms, denoted by "(div)," all scored higher than the non-diversity, "(non-div)," algorithms in unexpectedness, novelty, diversity, and the primary measure of serendipity. The two non-diversity based algorithms; however, both scored higher than the other three algorithms in successfulness and commonality. A one-way analysis of variance (ANOVA) with repeated measures showed that statistically significant differences among the means of the five

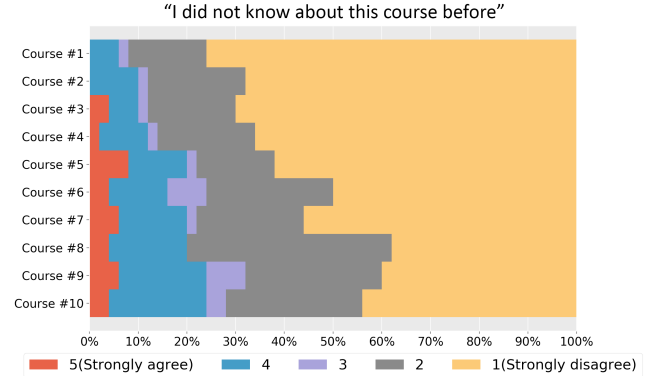
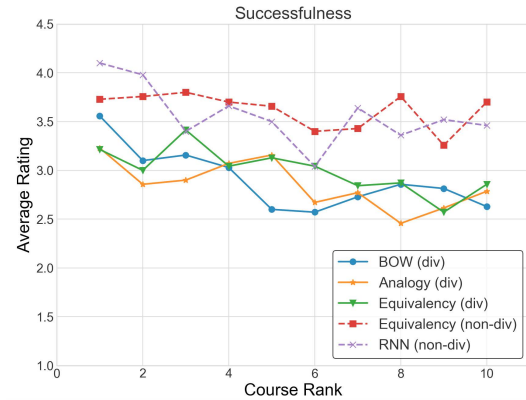
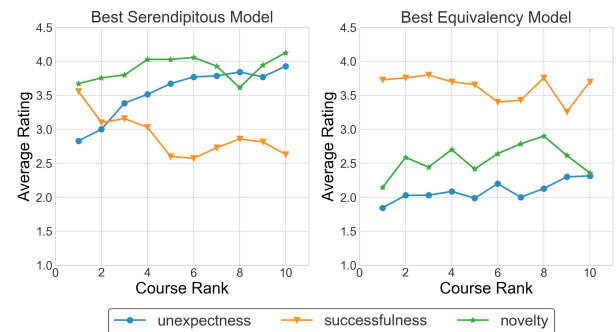
Table 3: (Online) Average student ratings of recommendations from the five algorithms in the six measurement categories

algorithm	unexpectedness	successfulness	serendipity	novelty	diversity	commonality
BOW (div)	3.550	2.904	3.227	3.896	4.229	3.229
Analogy (div)	3.473	2.851	3.162	3.310	4.286	2.986
Equivalency (div)	3.297	2.999	3.148	3.323	4.214	3.257
Equivalency (non-div)	2.091	3.619	2.855	2.559	2.457	4.500
RNN (non-div)	2.184	3.566	2.875	1.824	3.160	4.140

**Figure 3: Student novelty rating proportions by course recommendation rank produced by BOW (div)**

algorithms exist in all six measurement categories. All pairwise differences between diversity and non-diversity algorithms in the six measurement categories were statistically significant at the $p < 0.001$ level, given a Bonferroni correction for post hoc multiple comparisons. Among the diversity algorithms, there were no statistically significant differences except for BOW scoring higher than Equivalency (div) in unexpectedness and scoring higher than both Equivalency (div) and Analogy (div) on novelty. Among the two non-diversity algorithms, the statistically significant differences were the RNN scoring higher on diversity and Equivalency (non-div) scoring higher on novelty. With respect to serendipity, the div algorithms scored substantially and significantly higher (3.148-3.227) than the non-div algorithms (2.875 & 2.855). Div and non-div algorithms had complementary strengths, with div algorithms strong in unexpectedness and non-div algorithms strong in successfulness; however, the statistically significantly higher serendipity scores for the div algorithms were likely due to non-div algorithms' having a more severe weakness in unexpectedness (2.184 & 2.091) than div algorithms' weakness in successfulness (2.851-2.999). All analyses were conducted on the ratings a student gave per algorithm and measure, averaged over the ten courses they were shown.

The most dramatic difference can be seen in the measure of novelty, where BOW (div) scored 3.896 and the system's existing RNN (non-div) scored 1.824, the lowest rating in the results matrix. The proportion of each rating level given to the two algorithms on this question is shown in Figures 3 and 4. Hypothetically, an algorithm that recommended randomly selected courses would score high in both novelty and unexpectedness, and thus it is critical to also weigh their ability to recommend courses that are also of interest

**Figure 4: Student novelty rating proportions by course recommendation rank produced by RNN (non-div)****Figure 5: Successfulness comparison****Figure 6: BOW (div) vs. Equivalency (non-div) comparison**

to students. Figure 5 shows successfulness ratings for each of the algorithms aggregated by rank of the course result. The non-div algorithms, shown with dotted lines, always perform as well or better than the div algorithms at every rank. The more steeply declining slope of the div algorithms depicts the increasing difficulty of finding courses of interest across different departments. The tension between the ability to recommend courses of interest that are also unexpected is shown in Figure 6, where the best serendipitous model, BOW (div), recommends a top course of higher successfulness than unexpectedness, with the two measures intersecting at rank 2 and diverging afterwards. The best equivalency model, combining course description tf-idf and course2vec (non-div), maintains high successfulness but also maintains low unexpectedness across the 10 course recommendation ranks.

Are more senior students less likely to rate courses as novel or unexpected, given they have been at the University longer and been exposed to more courses? Among our sophomore (27), junior (22), and senior (21) level students, there were no statistically significant differences in novelty or unexpectedness ratings.

6.3 Qualitative Characterization of Algorithms

In this section, we attempt to synthesize frequent characterizations of the different algorithms shared by at least 10% of students (7) by looking at the answers they gave to the general feedback and observed commonality open response questions.

6.3.1 BOW (div). Several students remarked positively about recommendations matching to the themes of "art, philosophy, and society" or "design" exhibited in their favorite course. The word "language" was mentioned by 14 of the 61 respondents answering the open response question. Most of these comments were negative, pointing out the limitations of similarity matching based solely on literal course description matching. The most common critique given in this category was of the foreign spoken language courses that showed up at the lower ranks when students specified a favorite course involving programming languages. Other students remarked at additional dissimilarity when specifying a favorite course related to cyber security, receiving financial security courses in the results.

6.3.2 Analogy (div). The word "interesting" appeared in seven of the 54 comments left by students to describe commonalities among the analogy validation optimized algorithm. This word was not among the top 10 most frequent words in any of the other four algorithms. Several students identified broad themes among the courses that matched to their favorite course, such as "identity" and "societal development." On the other end of the spectrum, one student remarked that the results "felt weird" and were only "vaguely relevant." Another student stated that, "the most interesting suggestion was the Introduction to Embedded Systems [course] which is just different enough from my favorite course that it's interesting but not too different that I am not interested," which poignantly articulates the crux of difficulty in striking a balance between interest and unexpectedness to achieve a serendipitous recommendation.

6.3.3 Equivalency (div). Many students (seven of the 55) remarked positively on the commonality of the results with themes of data

exhibited by their favorite course (in most cases STATS C8, an introductory data science course). They mentioned how the courses all involved "interacting with data in different social, economic, and psychological contexts" and "data analysis with different applications." One student remarked on the algorithm's tendency to match around the main topic of the favorite course, saying "they were relevant if looking for a class tangentially related."

6.3.4 Equivalency (non-div). This algorithm was the same as the above, except that it did not limit results to one course per department. Because of this lack of department filter, 15 of the 68 students submitting open text responses to the question of commonality pointed out that the courses returned were all from the same department. Since this model scored highest on a validation task of matching to a credit equivalent course pair (typically in the same department), it is not surprising that students observed that results from this algorithm tended to come from the department of the favorite course, which also put them close to their nexus of interest.

6.3.5 RNN (non-div). The RNN scored lowest in novelty, significantly lower than the other non-div algorithm, and was not significantly different from the other non-div algorithm in successfulness. In this case, what is the possible utility of the collaborative-based RNN over the non-div Equivalency model? Many of the 47 (of 50) student answers to the open response commonality question remarked that the recommendations related to their major (mentioned by 21 students) and contained courses that fulfilled a requirement (mentioned by seven) as the distinguishing signature of this algorithm. Since the RNN is based on normative next course enrollment behavior, it is reasonable that it suggested many courses that satisfy an unmet requirement. This algorithm's ability to predict student enrollments accurately became a detriment to some, as seven remarked that it was recommending courses that they were currently enrolled in. Due to the institutional data refresh schedule, student current enrollments are not known until after the add/drop deadline. This may be a shortcoming that can be rectified in the future.

7 FEATURE RE-DESIGN

As a result of the feedback received from the user study, we worked with campus to pull down real-time information on student requirement satisfaction from the Academic Plan Review module of the PeopleSoft Student Information System. We re-framed the RNN feature as a "Requirements" satisfying feature that, upon log-in, shows students their personalized list of unsatisfied requirements (Figure 7). After selecting a requirement category to satisfy, the system displays courses which satisfy the selected requirement and are offered in the target semester. The list of courses is sorted by the RNN to represent the probability that students like them will take the class. This provides a signal to the student of what the normative course taking behavior is in the context of requirement satisfaction. For serendipitous suggestions, we created a separate "Explore" tab (Figure 8) using the BOW (div) model to surface the top five courses similar across departments, due to its strong serendipitous and novelty ratings. The Equivalency (non-div) model was used to display an additional most similar five courses within the same department. This model was chosen due to its strong successfulness ratings.

CCN	Class	Subject
19319	Social Implications of Computer Technology (195)	Computer Science
32560	Introduction to Science, Society, and Ethics (100G)	Interdisciplinary Studies
29166	Energy and Society (C184)	Public Policy
29166	Energy and Society (C100)	Energy and Resources
27356	Ethics in Science and Engineering (100)	Bioengineering

Figure 7: The “Requirements” Interface

8 DISCUSSION

Surfacing courses that are of interest but not known before means expanding a student’s knowledge and understanding of the University’s offerings. As students are exposed to courses that veer further from their home department and nexus of understanding, recommendations become less familiar with descriptions that are harder to connect with. This underscores the difficulty of producing an unexpected but successful course suggestion. Are recommendations useful when they suggest something expected or already known? Two distinct sets of responses to this question emerged from student answers to the last open ended feedback question. One remark stated, “The best algorithms were the ones that had more diverse options, while still staying true to the core function of the class I was searching. The algorithms that returned classes that were my major requirements/in the same department weren’t as helpful because I already knew of their existence as electives I could be taking.” While a different common view was expressed with, “I think the fifth algorithm [RNN] was the best fit for me because my major is pretty standardized.” These two comments make a case for both capabilities being of importance. When pluralism and discovery are priorities, recommender systems in higher education should provide services that optimize for serendipity, allowing learners to explore the terrain of courses or resources without a strong bias towards the student’s past experience and historic pathways. This approach caters to learners who are ready to pave their own path. When degree completion is the focus or learners need more guidance, student comments suggest that optimizing for successfulness using content-based or collaborative-based algorithms combined with course requirement filters is most sensible. A blend of the two may be worth pursuing in future work, where students are

Considerations across campus

- #1 Data Science Connector**
Letters & Science (88)
Connector courses are intended to connect the Foundations of Data Science (COMPSCI C8/INFO C8/STAT C8) course with particular fields of study. They will apply the concepts and techniques of the foundation course to topics of interest in a particular discipline in order for students to develop critical thinking in data in subject areas that most interest them; these courses also provide a more nuanced understanding of the context in which the data comes into existence.
- #2 Introduction to Urban Data Analytics**
City & Regional Planning (101)
This course (1) provides a basic intro to census and economic data collection, processing, and analysis; (2) surveys forecasting and modeling techniques in planning; (3) demonstrates the uses of real-time urban data and analytics; and (4) provides a socio-economic-political context for the smart cities movement, focusing on data ethics and governance.
- #3 Introduction to Ecological Data Analysis**
Env Sci. Policy, & Mgmt (173)
Introduces concepts and methods for practical analysis of data from ecology and related disciplines. Topics include data summaries, distributions, and probability; comparison of data groups using t-tests and analysis of variance; comparison of multi-factor groups using analysis of variance; evaluation of continuous relationships between variables using regression and correlation; and a glimpse at more advanced topics. In computer laboratories, students put concepts into practice and interpret results.
- #4 Cartographic Representation**
Geography (183)
Problems in the representation of quantitative and qualitative data on thematic maps.
- #5 The Person in Big Data**
Psychology (7)
This course will introduce students to the basic principles and methods of personality and social psychology as applied to a rapidly growing topic of modern society—the collection and analysis of online social “big data.” Students will learn about the ways in which big data has historically been defined, collected, and utilized, as well as fundamental concepts in person perception and social behavior that are relevant to topics of big data collection, analysis, and interpretation.

Matches within Sociology

- #1 The Power of Numbers: Quantitative Data in Social Sciences**
Sociology (7)
This course will provide students with a set of skills to understand, evaluate, use, and produce quantitative data about the social world. It is intended specifically for social science majors, and focuses on social science questions. Students will learn to: produce basic graphs, find good-quality and relevant data on the web, manipulate data in a spreadsheet, including producing pivot tables, understand and calculate basic statistical measures of central tendency, variation, and correlation, understand and apply basic concepts of sampling and selection, and recognize an impossible statistic.
- #2 Research Design and Sociological Methods**
Sociology (105)
Problems of research design, measurement, and data collection, processing, and analysis will be considered. Attention will be given to both qualitative and quantitative studies.
- #3 Popular Culture**
Sociology (163)
This course considers the relations between sociology and moral philosophy through an examination of classical and contemporary studies in both fields.
- #4 Virtual Communities/Social Media**
Sociology (167)
With the advent of virtual communities and online social networks, old questions about the meaning of human social behavior have taken on renewed significance. Using a variety of online social media simultaneously, and drawing upon theoretical literature in a variety of disciplines, this course delves into discourse about community across disciplines. This course will enable students to establish both theoretical and experiential foundations for making decisions and judgments regarding the relations between mediated communication and human community.
- #5 Introduction to Sociology**
Sociology (1)
Introduces students who are considering majoring in sociology to the basic topics, concepts, and principles of the discipline. This course is required for the major; 1 or any version of 3 is prerequisite for other sociology classes; students not considering a sociology major are directed to any version of 3.

Figure 8: The “Explore” Interface

encouraged to explore but within the the scope of courses likely to keep them progressing towards degree completion.

9 LIMITATIONS

The more distal a course description is, even if conceptually similar, the less a student may be able to recognize the commonality with a favorite course. A limitation of our study in demonstrating the utility of the neural embedding is that students had to rely on the course description semantics in order to familiarize themselves with the suggested course and determine if they were interested in taking it. If a concept was detected by the neural embedding but not the BOW, this likely meant that the concept was difficult to pick-up from the course description displayed to students. Past work has shown that users evaluate media recommendations less favorably before they take the recommendation than after when important aspects of the recommended content is not described in the recommendation [24]. Future work could augment recommended course descriptions with additional information, including latent semantics inferred from enrollments [11] or additional semantics retrieved from available course syllabi.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 1446641.

REFERENCES

- [1] Zeinab Abbassi, Sihem Amer-Yahia, Laks VS Lakshmanan, Sergei Vassilvitskii, and Cong Yu. 2009. Getting recommender systems to think outside the box. In *Proceedings of the third ACM conference on Recommender systems*. ACM, 285–288.
- [2] Andrew Abbott. 1999. Department and discipline. *Chicago sociology at one hundred* (1999).
- [3] Panagiotis Adamopoulos and Alexander Tuzhilin. 2015. On unexpectedness in recommender systems: Or how to better expect the unexpected. *ACM Transactions on Intelligent Systems and Technology (TIST)* 5, 4 (2015), 54.
- [4] Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1. 238–247.
- [5] Marco Baroni and Alessandro Lenci. 2010. Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics* 36, 4 (2010), 673–721.
- [6] Sorathan Chaturapruek, Thomas Dee, Ramesh Johari, René Kizilcec, and Mitchell Stevens. 2018. How a data-driven course planning tool affects college students' GPA: evidence from two field experiments. (2018).
- [7] Bodong Chen and Haiyi Zhu. 2019. Towards Value-Sensitive Learning Analytics Design. In *Proceedings of the 9th International Conference on Learning Analytics and Knowledge*. ACM, 343–352.
- [8] Hung-Hsuan Chen. 2018. Behavior2Vec: Generating Distributed Representations of Users' Behaviors on Products for Recommender Systems. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 12, 4 (2018), 43.
- [9] D Manning Christopher, Raghavan Prabhakar, and Schacetzal Hinrich. 2008. Introduction to information retrieval. *An Introduction To Information Retrieval* 151, 177 (2008), 5.
- [10] Martin Dillon. 1983. Introduction to modern information retrieval: G. Salton and M. McGill. McGraw-Hill, New York (1983). 448 pp., ISBN 0-07-054484-0.
- [11] Matthew Dong, Run Yu, and Zach A. Pardos. 2019. Design and Deployment of a Better Course Search Tool: Inferring latent keywords from enrollment networks. In *Proceedings of the 14th European Conference on Technology Enhanced Learning*. Springer, 480–494.
- [12] Rosta Farzan and Peter Brusilovsky. 2011. Encouraging user participation in a course recommender system: An impact on user behavior. *Computers in Human Behavior* 27, 1 (2011), 276–284.
- [13] Li Fei-Fei, Rob Fergus, and Pietro Perona. 2006. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence* 28, 4 (2006), 594–611.
- [14] Mouzhi Ge, Carla Delgado-Battenfeld, and Dietmar Jannach. 2010. Beyond accuracy: evaluating recommender systems by coverage and serendipity. In *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 257–260.
- [15] Balázs Hidasi, Massimo Quadrana, Alexandros Karatzoglou, and Domonkos Tikk. 2016. Parallel recurrent neural network architectures for feature-rich session-based recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 241–248.
- [16] Leo Iaquina, Marco De Gemmis, Pasquale Lops, Giovanni Semeraro, Michele Filannino, and Piero Molino. 2008. Introducing serendipity in a content-based recommender system. In *Hybrid Intelligent Systems, 2008. HIS'08. Eighth International Conference on*. IEEE, 168–173.
- [17] Weijie Jiang and Zachary A Pardos. 2019. Time slice imputation for personalized goal-based recommendation in higher education. In *Proceedings of the 13th ACM Conference on Recommender Systems*. ACM, 506–510.
- [18] Weijie Jiang, Zachary A Pardos, and Qiang Wei. 2019. Goal-based course recommendation. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*. ACM, 36–45.
- [19] Noriaki Kawamae, Hitoshi Sakano, and Takeshi Yamada. 2009. Personalized recommendation based on the personal innovator degree. In *Proceedings of the third ACM conference on Recommender systems*. ACM, 329–332.
- [20] Judy Kay. 2000. Stereotypes, student models and scrutability. In *International Conference on Intelligent Tutoring Systems*. Springer, 19–30.
- [21] Lili Kotlerman, Ido Dagan, Idan Szpektor, and Maayan Zhitomirsky-Geffet. 2010. Directional distributional similarity for lexical inference. *Natural Language Engineering* 16, 4 (2010), 359–389.
- [22] Omer Levy and Yoav Goldberg. 2014. Linguistic regularities in sparse and explicit word representations. In *Proceedings of the eighteenth conference on computational natural language learning*. 171–180.
- [23] Dekang Lin et al. 1998. An information-theoretic definition of similarity.. In *Icml*, Vol. 98. Citeseer, 296–304.
- [24] Benedikt Loepp, Tim Donkers, Timm Kleemann, and Jürgen Ziegler. 2018. Impact of item consumption on assessment of recommendations in user studies. In *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM, 49–53.
- [25] Sean M McNee, John Riedl, and Joseph A Konstan. 2006. Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *CHI'06 extended abstracts on Human factors in computing systems*. ACM, 1097–1101.
- [26] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- [27] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.
- [28] Tien T Nguyen, Pik-Mai Hui, F Maxwell Harper, Loren Terveen, and Joseph A Konstan. 2014. Exploring the filter bubble: the effect of using recommender systems on content diversity. In *Proceedings of the 23rd international conference on World wide web*. ACM, 677–686.
- [29] Aditya Parameswaran, Petros Venetis, and Hector Garcia-Molina. 2011. Recommendation systems with complex constraints: A course recommendation perspective. *ACM Transactions on Information Systems (TOIS)* 29, 4 (2011), 20.
- [30] Zachary A Pardos, Zihao Fan, and Weijie Jiang. 2019. Connectionist recommendation in the wild: on the utility and scrutability of neural networks for personalized course guidance. *User Model User-Adap Inter* 29, 2 (2019), 487–525.
- [31] Zachary A Pardos and Andrew Joo Hun Nam. 2018. A Map of Knowledge. *CoRR preprint, abs/1811.07974* (2018). <https://arxiv.org/abs/1811.07974>
- [32] Agoritsa Polyzou, N Athanasios, and George Karypis. 2019. Scholars Walk: A Markov Chain Framework for Course Recommendation. In *Proceedings of the 12th International Conference on Educational Data Mining*. 396–401.
- [33] Zhiyun Ren, Xia Ning, Andrew S Lan, and Huzefa Rangwala. 2019. Grade Prediction Based on Cumulative Knowledge and Co-taken Courses. In *Proceedings of the 12th International Conference on Educational Data Mining*. 158–167.
- [34] Leonardo FR Ribeiro, Pedro HP Saverese, and Daniel R Figueiredo. 2017. struc2vec: Learning node representations from structural identity. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 385–394.
- [35] Guy Shani and Asela Gunawardana. 2011. Evaluating recommendation systems. In *Recommender systems handbook*. Springer, 257–297.
- [36] Mitchell L Stevens and Ben Gebre-Medhin. 2016. Association, service, market: Higher education in American political development. *Annual Review of Sociology* 42 (2016), 121–142.
- [37] Peter D Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research* 37 (2010), 141–188.
- [38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.
- [39] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*. 3630–3638.
- [40] Yuan Cao Zhang, Diarmuid Ó Séaghdha, Daniele Quercia, and Tamas Jambor. 2012. Auralist: introducing serendipity into music recommendation. In *Proceedings of the fifth ACM International conference on Web Search and Data Mining*. ACM, 13–22.