Data-Assistive Course-to-Course Articulation Using Machine Translation

Zachary A. Pardos
UC Berkeley
Berkeley, CA, USA
zp@berkeley.edu

Hung Chau
University of Pittsburgh
Pittsburgh, PA, USA
hkc6@pitt.edu

Haocheng Zhao
UC Berkeley
Berkeley, CA, USA
zhc@berkeley.edu

ABSTRACT

Higher education at scale, such as in the California public post-secondary system, has promoted upward socioeconomic mobility by supporting student transfer from 2-year community colleges to 4-year degree granting universities. Among the barriers to transfer is earning enough credit at 2-year institutions that qualify for the transfer credit required by 4-year degree programs. Defining which course at one institution will count as credit for an equivalent course at another institution is called course articulation, and it is an intractable task when attempting to manually articulate every set of courses at every institution with one another. In this paper, we present a methodology towards making tractable this process of defining and maintaining articulations by leveraging the information contained within historic enrollment patterns and course catalog descriptions. We provide a proof-of-concept analysis using data from a 4-year and 2-year institution to predict articulation pairs between them, produced from machine translation models and validated by a set of 65 institutionally pre-established course-to-course articulations. Finally, we create a report of proposed articulations for consumption by the institutions and close with a discussion of limitations and the challenges to adoption.

Author Keywords

Higher education, course-to-course articulation, machine translation, enrollment data, credit mobility

INTRODUCTION

Course articulation has been the bridge that connects programs from different levels of higher education to one another, forming pathways to achievement focused on equity of access. Across the United States, there is evidence that these pathways have been underperforming. Around 45% of the 20 million students entering higher education in the United States begin their post-secondary experience at 2-year public institutions [4]. A 2010 US Department of Education survey of 19,000 "Beginning Postsecondary Students" (BPS) found that 81.4% of community college students had aspirations of transferring to earn a 4-year degree [12]. Data on 852,439 public community college students, collected by the National Student Clearinghouse (NSC); however, found that only 13% had earned a 4-year degree in six years after beginning at a community college [14]. The picture looks better for those in the study who successfully transferred, with 42% of these students having completed their 4-year degree. Course articulation, or a lack of it, is not the primary culprit for these low outcomes; however, it is likely not an insignificant source either. Evidence of this is an analysis of the BPS data from the US Government Accounting Office's (GAO) in which it is estimated that 42% of credit earned at the community college level is lost upon transfer to a 4-year institution [3]. Much of this loss is due to switching majors or earning an excess of general credit before declaring a major, though it is estimated that a portion is due to lack of articulation and that a 20% increase in 4-year degree attainment, among transfers, can be expected if those articulations existed [9]. The impact of insufficient articulation on student rates of successful transfer has not been quantified, but a recent spate of state and national efforts to define additional pathways suggest that it has been an important factor [1, 5, 13]. These observations serve as mounting evidence that providing more comprehensive articulations can help improve transfer success through greater credit mobility.

Articulations at the degree level are often created by state mandate, with courses being developed at the 2-year and 4-year institutions in unison, or one modeled after the other, and with collaboration between faculty at both. Outside of these degree level articulations are those made on a course-by-course basis. In this case, there is a significant bottleneck of human resources committed to processing and validating requests for articulation. Each campus typically has a designated articulation officer, or chief instructional officer [15]. This person is responsible for receiving articulation requests, choosing which to consider, and then beginning the process of validation by conferring with the instructor of record at the other institution by way of its respective articulation officer. A diagram of the articulation process in the California public post-secondary system [2] is depicted in Figure 1. If only considering articulation of courses from one of the 115 2-year California Community Colleges (CC) to courses at the nine 4-year University of California (UC) campuses, there are 63M pairs (1,000*7,000) to consider, assuming[1] a catalog of 1,000 courses at the CC

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, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

L@S'19, June 24–25, 2019, Chicago, IL, USA

© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ISBN 978-1-4503-2138-9…$15.00

DOI: 10.1145/3330430.3333622

[1]These numbers are assumed based on counts extrapolated from our dataset consisting of enrollments from one UC and one CC

RIGHTSLINK

and 7,000 at each UC. This number can be reduced if assuming there always exists a clear department-to-department mapping between institutions, which is not always the case, and that courses are only considered for articulation within the mapped-to department. In this case, the lower bound number of course pairs to consider is 35,000 (20*35*50) assuming an average of 20 courses per department at the CC, 35 courses per department at the UC, and 50 departments at the CC articulating only to a single respective department at the UC. This number increases significantly when considering and maintaining articulation to the 23 institutions in California's State University System and articulation to the other 114 community colleges, necessary for lateral (CC-to-CC) transfer. The intractability of effectively curating an articulation database with a manual process increases exponentially when considering articulation to out-of-state or private institutions. The GAO estimates that 94% of credits are lost when transferring from a public to private institution [3].

In this paper, we posit that institutional big data have been an underutilized source that can be leveraged towards combating the bleak combinatorics of course-to-course articulation. We investigate the utility of these sources using the two datasets of course enrollments and course descriptions, one from a 4-year University of California campus (referred to as UC1) and one from a 2-year California Community College (referred to as CC1). Recent work has found that analysis of enrollment sequences using word2vec approaches can embed courses into a space of semantic structure [10] similar to the space words are embedded into based on their word contexts in a corpus [6]. We build on this finding to test if a translation can be learned between the course spaces of two different institutions, just as it has been learned between the word spaces of two different languages [7].

DATASETS

UC1 dataset

Our UC1 dataset consists of 7,487 courses in 179 departments taken between 2008 and 2017 at the Berkeley campus of the University of California. We inherit pre-trained vectors for each course from the authors of prior work [10]. These continuous valued vectors are 300 dimensions in length trained from 4.8 million enrollments from 164,196 students using a skip-gram model and tested against a validation set of within-institution course credit restrictions (i.e., equivalencies) curated by the university. Details of this training is explained later in the models section of this paper. Also found in this dataset are the plain-text catalog description of each course. The average length of a UC1 course description is 325 words and there are 489 descriptions with fewer than 10 words.

CC1 dataset

Our CC1 dataset consists of 1,000 courses in 53 departments taken between 2013 and 2018 at Laney Community College of Oakland, located six miles south of UC Berkeley. This is a novel dataset for which no prior models had been trained. The average length of CC1 course descriptions is 27 and there are 62 descriptions which have less than 10 words. Additionally, this dataset contains 298,174 enrollments, and their semester and year, made by 58,716 students.

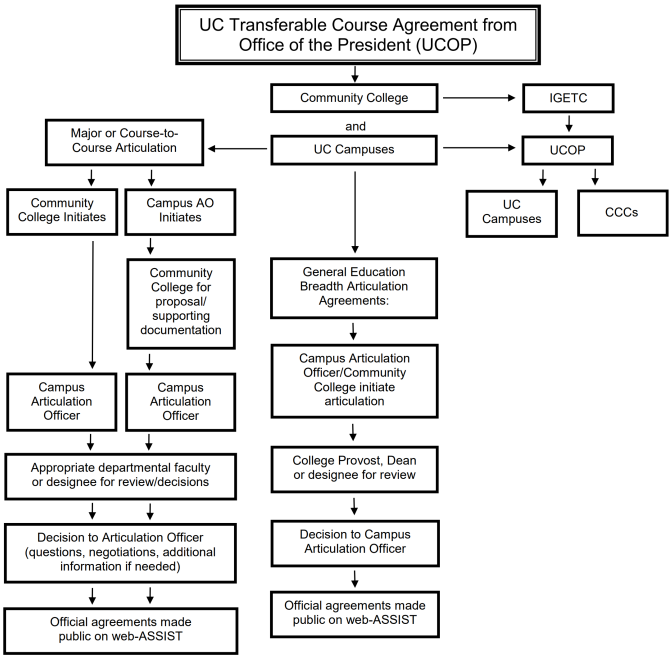

Figure 1. Diagram of the process for course articulation in the University of California system, sourced directly from the California Articulation Policies and Procedures Handbook. The process for course-to-course articulation can be seen by following the left side of the flow diagram.

UC1's course	*CC1's course(s)*
AFRICAM5B	AFRAM_31
ASAMST20A	ASAME_45A; ASAME_45B
ASAMST20C	NO COURSE ARTICULATED

Table 1. Course articulation samples from assist.org. Multiple CC1 courses denote that both must be taken to count towards the UC1 course credit.

Validation set

We use the existing set of course articulations between UC1 and CC1 to evaluate the predicted articulations of our models. These articulation pairs were screen scraped and manually enumerated from assist.org[2], the official information system for looking-up articulations within the California public post-secondary system. The system lists the articulations that exist between the two institutions with respect to each major offered at CC1. The total number of articulation pairs extracted was 65. Given our goal of proposing new potential articulations, we also curated a list of major satisfying courses at UC1 for which there were no respective articulated courses at CC1. There were 184 such UC1 courses. Table 1 shows samples from this course articulation dataset.

MODELS

In this section, we present several models for course representation from which to predict the similarity between courses at the two institutions. We will also describe the application of machine translation as a linear transformation from a source course vector space (i.e., UC1) to target course vector space

[2]These articulations were kept current up until 2017. A new system, with updated articulations, is expected within the year.

RIGHTSLINK

(i.e., CC1). This technique is applied to our course2vec based course representations.

Collaborative-based model (course2vec)

We use an adaptation of word2vec applied to course enrollment sequences as described in prior work [10, 11]. The data are prepared by enumerating course enrollment sequences per student with the enrollment sequence consisting of course ID tokens (e.g., ECON_141) sequenced in the order in which the student took the courses. Courses taken in the same semester are serialized by randomizing their within-semester order. A skip-gram is then applied to these sequences exactly as it would be applied to sequences (or sentences) of words in a language context to produce continuous vectors for each course. Prior work has found that these vectors, learned from enrollment sequences, encode information about the topic of the course, as well as latent attributes such as its mathematical rigor and the most common major of students taking the course [11]. For the UC1 dataset, these vectors were pre-trained and inherited from the authors of that prior work. For CC1, we train course vectors, sweeping the hyper-parameters of vector size and window size and perform model selection based on the leave-one-out predictive performance on our articulation validation set, described in detail in a later section. There is a threat of overfit in this approach; however, we consider it to be minor given course2vec is an unsupervised process and a limited number of hyper-parameter combinations are used with which to generate candidates for model selection.

Content-based models

Course catalog description is the source of similarity data used by this class of models. We consider three different course representations utilizing these data; simple bag-of-words, tf-idf, and an average of the respective word vectors of words in the description using a pre-trained word embedding.

BOW with term-frequency

In our simple bag-of-words (BOW) model, each course is represented as a vector of the length of the total unique words in all courses across both UC1 and CC1. The values in a course's vector are *zeros* unless the word of the corresponding position in the vector has occurred in the description, in which case the frequency of this word in the description is used. Similarity between courses can be calculated using cosine similarity of their respective BOW. We applied a few filters to course descriptions before constructing the BOW of courses for both institutions. First, was to filter out non-words (e.g., course numbers) from the descriptions. Second, we removed the top 100 most frequent words (e.g., course, student, credit) from all descriptions. After filtering, we were left with a vocabulary of 14,316 across all descriptions.

TF-IDF

The simple BOW model assigns word frequencies as weights to words. However, if a word appears frequently in most of the courses, it will not help to differentiate between courses, nor help in identifying which are truly similar. We consider a *tf-idf* (term frequency-inverse document frequency) representation to address this issue, assigning a weight to a particular term t in a course description d. As a result, instead of representing a course description as a vector of *word frequencies*, each dimension of the vector is a real-valued *tf-idf* weight ($w_{t,d}$), calculated as following:

$$w_{t,d} = (1 + log(tf_{t,d})) \times log_{10} \frac{N}{df_t} \quad (1)$$

in which,

- $tf_{t,d}$: frequency of term t in course description d
- df_t: number of course descriptions in which term t appears
- N: number of course descriptions in the collection

Word2vec (DescVec)

There is a large disparity in the length of descriptions between UC1 (325 words) and CC1 (27 words). Anticipating that this may introduce noise into the process of finding similar courses based on description, we decided to attempt to ameliorate this issue by representing both institution's course descriptions using a pre-trained public word embedding provided by the seminal word2vec work [6, 8]. This embedding was trained on 100 billion words from Google News with a vector size of 300 dimensions. To represent course vectors using this word embedding, we average over all the vectors of the words appearing in the course descriptions after applying the same pre-processing as described in the above BOW sections. This approach, which we refer to as *DescVec*, is anticipated to have the added benefit of still finding similarity between courses if they use different, but synonymous words.

Model combination

The content-based and collaborative-based course representations are produced from entirely different sources of information about courses and are likely to poss their own benefits and deficits. The content-based models represent the content of the course as described by the instructor; however, the description can become out of date and a course can be described in an overly brief or generic way. In comparison, the course2vec models use student enrollment behaviors to inform the representation of a course. Because of this, they may contain important information about the course known by students (e.g., which are the courses with a reputation for being easy) but not expressed by the instructor in the description. Conversely, course2vec representations may suffer from noise in the case of courses with low enrollment (a minimum enrollment of 15 was set in the model) and also will suffer from courses that have recently changed their content considerably from historic offerings. In order to allow these two models to contribute their complementary benefits, we add a model to our evaluations which is a combination of the DescVec model and the course2vec model.The combining process is as follows:

1. *Source course vector concatenation*. Firstly, the source course vectors are transformed to the target course embedding space through a machine translating process detailed in the next section. We then concatenate the translated source course vectors with their respective DescVec course vectors (see the upper concatenation in Figure. 2).

RIGHTSLINK

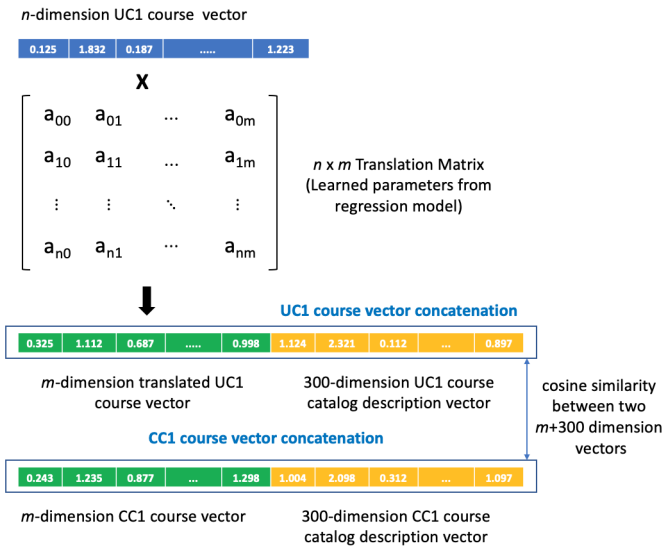

Figure 2. Process of translating a UC1 *course2vec* vector to the CC1 space and concatenating it with its *DescVec* vector for matching to a concatenated CC1 course vector via cosine similarity.

2. *Target course vector concatenation*. No translation is needed. We only concatenate the target course vector with its respective DescVec course vector (see the lower concatenation in Figure. 2).

Machine Translation

For the course2vec model, UC1 course vector set and CC1 course vector set are learned separately; thus, they do not share the same coordinate frame of reference and their embeddings are subsequently different. Moreover, the dimensions of the two vector spaces are not the same. We can not directly calculate the similarity between two course vectors coming from two different spaces. However, a linear translation between skip-gram embeddings can be learned as demonstrated by Mikolov at el. [7] that showed that the same concepts (e.g., *animals*) in different languages have similar relative geometric arrangements in their embeddings. By applying the linear translation of scaling and rotation, a reasonable mapping between the two language spaces could be found based on a small set of preexisting word translation pairs. This is the key idea behind the parallel we draw to course embedding translation where we base the learning of this translation of two institution embedding spaces on a small set of preexisting course articulation pairs.

Regression-based translation

Since we anticipate that the same courses in different institutions are likely to have similar geometric arrangements in their respective institution's embeddings, the transformation from one vector embedding space to another can be expected to be linear. We perform a general linear regression with the input vector $\mathbf{s} \in \mathbb{R}^n$ and the output vector $\mathbf{t} \in \mathbb{R}^m$, in which n and m are the sizes of the dimensions of the source vector space and target vector space, respectively. The goal of our model is to minimize the differences between the *translated source course vectors* and *target course vectors* in the N articulation pairs. The optimization problem is described as follows:

$$min_{trans} \sum_{i=1}^{N} dist(trans(s_i), t_i) \quad (2)$$

The function ***trans*** is used to translate a course vector from the source embedding space to the target embedding space using the optimized weights $\boldsymbol{W}$ and biases $\boldsymbol{b}$ (also called translation matrix $\boldsymbol{M}$) obtained from the regression model. The ***dist*** function is the *loss* function in the regression model. We use *cosine_proximity* and *mse* loss functions to train our models, discussed more in section "Cosine vs Euclidean". Stochastic gradient descent is used as the optimizer to fit the model to our data. After translating a course vectors from the source embedding space to the target embedding space, the translated course vector now has the same number of dimensions as all the target course vectors, allowing it to be compared with target course vectors using metrics such as cosine similarity and Euclidean distance.

Articulation Prediction

The goal of our methodology is that, given a course c in one institute, we would like to predict an ordered list of courses in another institute that are most similar to c. With the course representations and machine translated vectors in-hand, we can compute the similarity or distance between two courses from different institutions. The course articulation process is described as following:

1. Represent all courses by one of the course representation models.
2. Translate the source course vector s through the machine translation process if the vector was produced by the course2vec or combined model. Otherwise, use the original representation of the source course vector (i.e., content-based models).
3. Compute the cosine similarity (Equation 3) or Euclidean distance (Equation 4) between the source course vector (or the translated source course vector) and all the course vectors in the target institute.
4. Rank the target institute courses based on their similarity or distance scores, and choose the top k (e.g., 10) courses for articulation recommendation.

EVALUATION

In this section, we discuss how we validate our models, which includes choosing the hyper parameters for CC1's course2vec model, choosing between cosine similarity and Euclidean to find similar courses, and considering the difference in performance of our articulation predictions if we limit predictions to a similar department at the target institution.

Since our validation set only contained 65 labelled articulation pairs, we use a leave-one-out cross-validation. We use the metric of recall @ k to evaluate prediction performance. This means that, for each of the 65 pairs, given the UC1 course

RIGHTSLINK

in the pair, we obtain the top k ranked CC1 courses from the articulation prediction process explained in the previous section. The recall is calculated based on the percentage of correct CC1 courses that fall within the top k. This metric was chosen because of the anticipated scenario where we generate an articulation report to the articulation officer of CC1. This report will not show just one suggested CC1 course per unarticulated UC1 course, but rather a list of suggestions. The k in recall @ k represents the length of this hypothetical list and the recall metric represents the percentage of the 65 lists of length k that included the true articulation(s) in them.

Parameter search

The two most crucial hyper-parameters of the skip gram model are vector size v and window size w. Modification of the vector size is a way to tune the granularity with which regularities are produced in the feature space. Different languages and dataset sizes will require different vector size settings to achieve the same granularity. It is desirable for the granularity of both course vector sets to be at the same level for the feature mapping to be effective. We therefore conduct a minimal hyper parameter search of the CC1 course2vec model. We start with the pre-trained course vectors from UC1. Then, we sweep a small range of vector sizes and window sizes for CC1's course2vec by optimizing the leave-one-out recall performance described in the above section. We chose recall @ 5 as the k used for optimization as this was an ad-hoc estimate for a reasonable length list of courses for an articulation officer to consider. This process of hyper-parameter tuning went as follow:

1. For each of the 27 parameter sets, we run course2vec 20 times to learn different CC1 course vectors based on different random model initializations
2. Obtain the average recall @ 5 from leave-one-out cross validation described above for each CC1 course vector set
3. Average over the recalls @ 5 of all the 20 course vector sets for particular parameter sets
4. Select the parameter set with the best average recall performance

The result from Figure 3 shows that, within the 27 parameter sets, the vector size 20 and window size 5 achieved the best perform w.r.t recall @ 5. Therefore, we chose these values to train the final CC1 course vector set.

Cosine vs Euclidean

Given vector-space course representations, if vectors come from the same original space, it is effective to directly calculate their distances using cosine or Euclidean distances. On the other hand, vectors coming from different spaces need to be mapped to the same space by the proposed method explained in Section "Regression-based translation". After the transformation, we can calculate their distances.

- *Cosine similarity:* measure the similarity between two non-zero vectors $\mathbf{x}$ and $\mathbf{y}$ by computing the cosine of the angle between them.

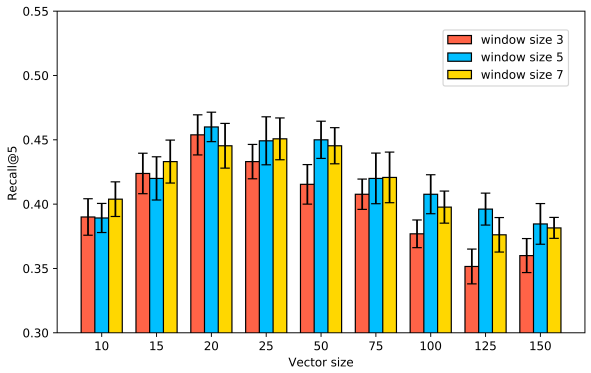

Figure 3. Recall performance @ 5 with different sets of course2vec vector sizes and window sizes for training CC1 vectors. The error bars represent 95% confidence intervals, obtained by running each model 20 times.

$$cosine\_similarity(\mathbf{x},\mathbf{y}) = \frac{\mathbf{x}.\mathbf{y}}{\|\mathbf{x}\|\,\|\mathbf{y}\|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\,\sqrt{\sum_{i=1}^{n} y_i^2}} \tag{3}$$

- *Euclidean distance:* measure the straight-line distance between two points in Euclidean space.

$$Euclidean\_distance(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{4}$$

Depending on which distance metric is used for evaluation, matching to an appropriate loss function used to optimize the problem defined in Equation 2 may be called for. If cosine similarity is used to evaluate, we can use cosine_proximity as the loss function, and mean squared error (mse) as the loss function in the case of Euclidean as the evaluation metric.

Department filtering

It is intuitive to think that course articulation pairs should be in equivalent departments across colleges (e.g., a course offered by Mathematics department at UC1 should be articulated to a course offered by Mathematics department at CC1). We therefore also compare the performance of the best model to its department filtering version. However, among the 65 articulation pairs, there are 2 pairs that come from departments that were not mapped to one another in the department mapping conducted by the authors. These were STAT2 to MATH13 and NUSCTX10 (Nutritional Sciences and Toxicology) to BIOL28 (Biology). In order to have a fair comparison for *with* and *without* department filtering data, we excluded these two pairs, leaving us with 63 articulation pairs for that evaluation.

RESULTS

In this section, we present the articulation prediction performance of the different models and present visualizations of the course embeddings for intuition. For the course2vec model and combined model, As we obtained from our development experiments, the performances of models trained with *mse*

RIGHTSLINK

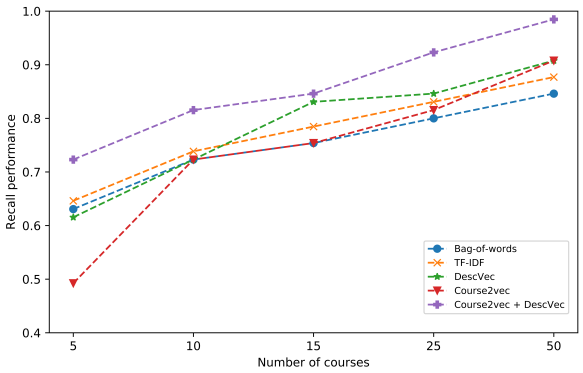

Figure 4. Recall comparison of the different models trained with *cosine_proximity* loss function @ k.

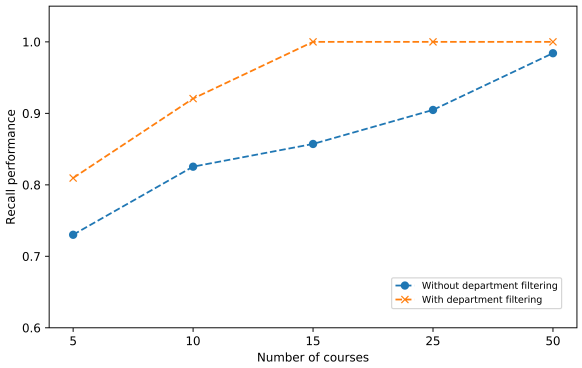

Figure 5. Recall comparison of the *CourseVec + DescVec* model with and without department filtering.

Course Representation	*Median Rank*	*Mean Rank*	*Std of Rank*
Bag of words	3.0	59.12	173.28
TF-IDF	3.0	57.01	177.65
DescVec	3.0	21.06	57.94
course2vec	6.0	17.74	33.65
course2vec+DescVec	**2.0**	**7.94**	**15.73**

Table 2. Course articulation ranking validation from the different course representations.

loss function were worse than the ones trained with *cosine* loss function. Therefore, we only report the results for the models trained with *cosine* loss functions (see Figure 4).

In addition to the recall performances @ k, we also report the rank of the true articulated course in our prediction results. The median and median rank across the 65 articulation predictions is reported, as well as the standard deviation of ranks (see Table 5).

Observations:

- Although slim, the BOW model with TF-IDF shows consistent improvement over the term-frequency BOW model across values of k.
- Among the content-based models, the DescVec model performs best, overall.
- The course2vec model performs substantially worse than the content-based models on recall @ 5 but then matches their performance for all other values of k. It also can be observed from Table 5 that, while having a higher median rank, the course2vec model's mean and std are lower than the content-based models, suggesting that it has fewer poor performing outliers.
- The combined model (course2vec + DescVec), which leveraged the strength of both the content-based and collaborative-based models, shows the best performance across all values of k and among all the rank metrics.

Figure 5 shows the difference in performance between the combined model with and without department filtering. The performance of the model with department filtering brings recall @ 5 up above 80%. An interpretation of this result is that if the model were to produce a set of five CC1 course articulation suggestions for each one of ten chosen UC1 courses, eight of those sets of ten suggestions can be expected to contain an appropriate articulation course.

Visual inspection of course2vec models

We visually inspect the CC1 and UC1 course2vec models to investigate if similar geometric regularities can be seen as they have been in visualization of language translation models [7]. We use PCA to reduce the course2vec vectors to 2-dimensions, then zoom into the the Computer Science departments of each visualization to compare the relative positions of courses with articulations between UC1 and CC1. As we can see from Figure 6, the 2D course vectors obtained from the skip-gram models show a similar, but not perfectly so, geometric arrangements of articulated courses. Computer Science was picked because courses within that department performed well on the articulation task and produced a PCA visualization that underscores why the course2vec representations learned from course enrollments alone can be effective.

Moreover, we also inspect the course representations for all courses at each institution, this time using t-Stochastic Neighborhood Embedding [16] to reduce the course2vec+DescVec representations to 2-dimensions. The t-SNE algorithm is chosen in this case because it is generally better at retaining global embedding structure than PCA. This visualization (Figure 7) reveals a few regularities in the relative department-level positions of the two institutions as well as a tight grouping of courses by department, also observed in [11]. The deparments of *Chemistry*, *Engineering*, *Civ Eng*, *Architecture*, *African American Studies*, and *American Studies* can be seen as appearing in clockwise order in both institutions, underscoring why the representations learned from course enrollments by course2vec alone, rival the information contained in course descriptions.

ARTICULATION SUGGESTION REPORT

The intent of this work is to lead to more articulations being produced across the system. In order to achieve this impact

RIGHTSLINK

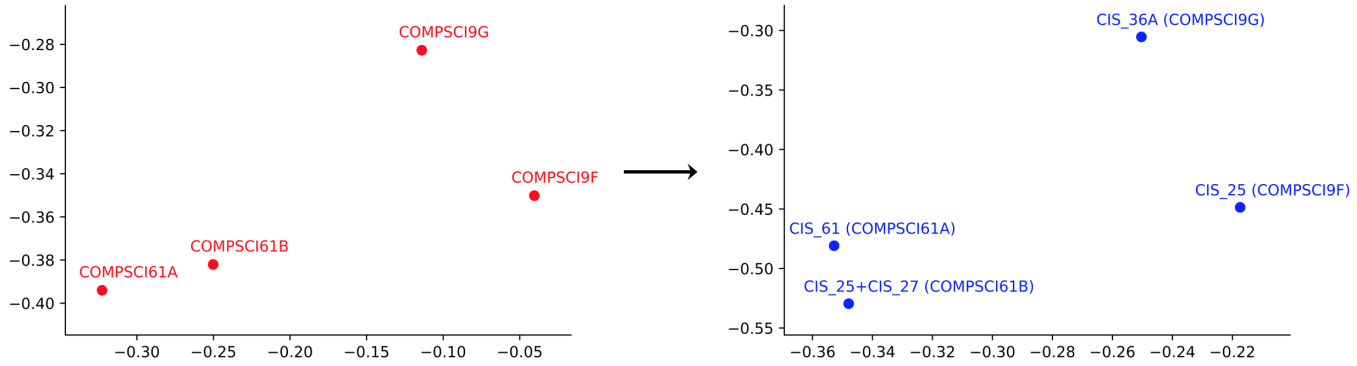

Figure 6. Distributed vector representations of Computer Science courses in UC1 and CC1. The four course vectors are reduced to two dimensions using PCA in each of the institutions. UC1 includes *Structure and Interpretation of Computer Programs* (COMPSCI61A), *Data Structures* (COMPSCI61B), *C++ for Programmers* (COMPSCI9F) and *JAVA for Programmers* (COMPSCI9G). CC1 includes *Structure and Interpretation of Computer Programs* (CIS_61), the combination of *Object Oriented Programming Using C++* (CIS_25) and *Data Structures and Algorithms* (CIS_27), *Object Oriented Programming Using C++* (CIS_25) and *Java Programming Language I* (CIS_36A).

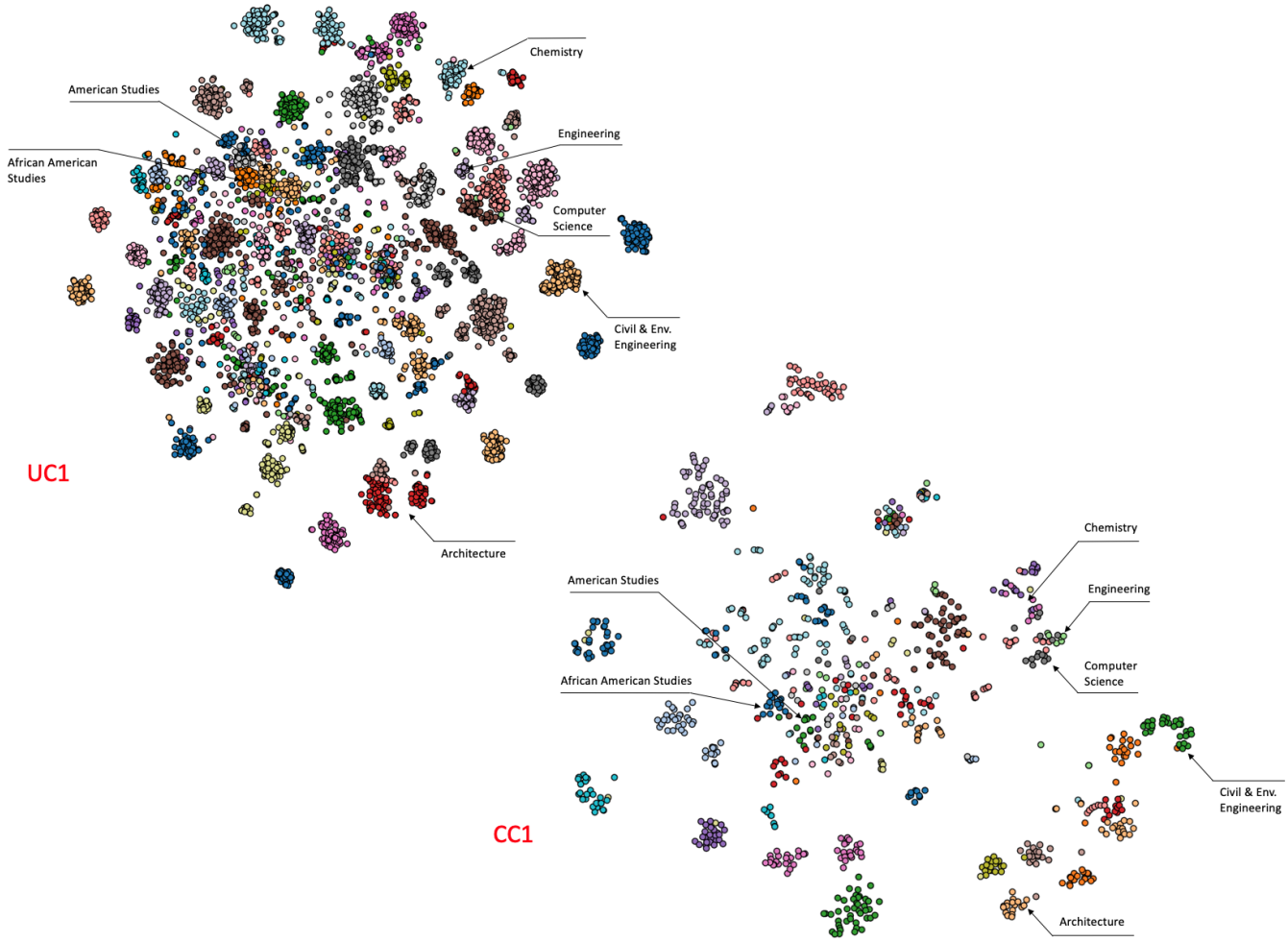

Figure 7. t-SNE scatter plots of courses obtained from the *course2vec+DescVec* models. The color of the points represent the departments and the text annotations represent the names of the departments which have sufficient courses and direct mappings between UC1 and CC1.

RIGHTSLINK

without an overhaul of the UCOP articulation process, we see this methodology as fitting into the process by producing an articulation suggestion report destined either for the articulation officer or direct to the faculty of suggested courses at CC1. In this section, we describe this last-leg of the process by considering the production of this report. There are 184 UC1 courses listed in assist.org that do not yet have CC1 courses articulated to them. All of these might be reasonable to provide suggestions for, but since faculty and the articulation officer can be expected to have limited time and resources to evaluate suggestions, we must consider a subset of the 184 to be privileged for inclusion in the report over the others and if a reasonable heuristic exists with which to choose these courses. All results in this section are based on the best performing representation, the *course2vec+DescVec* model with cosine_proximity loss.

Heuristics for choosing candidate UC1 courses

Among the 184 UC1 courses that do not have articulated courses at CC1, we filter out courses from UC1 departments that do not exist at CC1, resulting in 155 remaining UC1 courses. While a subset of these courses could be selected at random for inclusion in the report, an improvement on random selection can be made if a heuristic of UC1 courses exists that correlated with CC1 articulation performance. This would allow us to put together a report most likely to lead to successful articulation. In order to prioritize UC1 courses for articulation suggestion, we explore heuristics based on the three following proposed metrics:

- **Metric 1**: correlation between '*the distance of predicted UC1 to CC1 articulation vector to the vector of the nearest CC1 course*' and '*the rank of the target CC1 course vector*'
- **Metric 2**: correlation between '*the distance of predicted UC1 to CC1 articulation vector to the target CC1 course vector*' and '*the rank of the target CC1 course vector*'
- **Metric 3**: correlation between '*the number of CC1 course vectors that fall within a threshold θ of the distance to the predicted UC1 to CC1 articulation vector*' and '*the rank of the target CC1 course vector*'. We define θ as the average distance between the predicted vectors and the target vectors in the validation set.

The intuition behind Metric 1 is that if the translation of the UC1 course vector into the CC1 space places it in the middle of nowhere, far from any CC1 courses, this may be a sign that it will produce poor suggestions. Metric 2 requires the information of the actual target course given as input, which we do not have for the 155 unarticulated UC1 courses; however, it serves as a valuable upper bound and a sanity check on the behavior of our proposed method. We expect that the distance of predicted vector to the target course vector should be highly correlated to the rank of the target. The intuition of Metric 3 is that the density of CC1 courses around the predicted vector may have a correspondence with the quality of suggestions, though we did not have a hypothesis if this corrrelation would be positive or negative. For all three metrics, we calculated correlations using the two different metrics of cosine similarity and Euclidean distance, trying mse and cosine_proximity loss with both. We also tried the course2vec+DescVec model

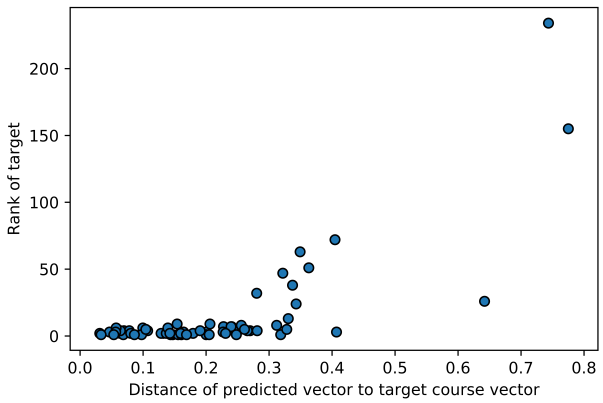

Figure 8. Demonstration of the correlation between '*the distance of predicted vector to the target course vector*' and '*the rank of the target*', Metric 2, obtained from the combined model with *cosine* distance and without department filtering.

both with and without department filtering. On the reference Metric 2, the model trained with *mse* loss without department filtering obtained a statistically significant correlation of *0.756* (depicted in Figure 8). Metric 1 scored the second highest correlation of *0.565* from the model trained with *mse* loss function without department filtering and using *cosine* distance. We therefore chose this model and heuristic for selecting courses for articulation suggestion.

Compiling the final report

Our goal was to produce an articulation report that took no longer than two hours for an articulation officer to evaluate. We anticipated that the clear false-positives, comprising the majority of suggestions, could be quickly dismissed and therefore estimated an average of 30 seconds to evaluate each suggestion. Based on results of the leave-one-out evaluation, a considerable bump in recall accuracy is observed when producing 10 or 15 suggestions instead of 5. Therefore, to increase the changes our suggestions contained the diamond in the rough, we chose to produce 15 CC1 suggestions for each UC1 course. In order to limit the total officer evaluating time to two hours, this meant choosing ten UC1 courses which would be based on their Metric 1 score. In the report, we provide the catalog description of the UC1 course as well as a link to its syllabus for quick reference. Catalog descriptions for the 15 suggested CC1 courses to articulate to are also included in the report. Tables 3 and 4 show sample courses from the report.

DISCUSSION

We found that a simple word2vec approach to articulation, *DescVec*, performed equal to or better than the experimental course2vec machine translation model. However, the experimental model was shown to contain novel useful information in addition to what was found in the course description based *DescVec* model. This was made evident by the performance of the concatenation of the course2vec model vectors with the *DescVec* vectors which performed meaningfully better than any other model in our recall @ k metric for all values of k. It also performed between 30 and 50% better than the second

RIGHTSLINK

Campus	*Course ID*	*Course Title*
UC1	ECON2	Introduction to Economics
CC1	ECON_1	MACRO-ECONOMICS
CC1	ECON_2	MICRO-ECONOMICS
CC1	BUS_1A	FINANCIAL ACCOUNTING
CC1	BUS_1B	MANAGERIAL ACCTG
CC1	MATH_16A	CALCULUS-BUS/SOCSC
CC1	BUS_2	INTRO TO BUS LAW
CC1	MATH_1	PRE-CALCULUS
CC1	BUS_10	INTRO TO BUSINESS
CC1	MATH_13	INTRO TO STATISTICS
CC1	MATH_16B	CALCULUS-BUS/SOCSC
CC1	BUS_21	PAYROLL ACCOUNTING
CC1	MATH_3A	CALCULUS 1
CC1	BUS_4	COST ACCOUNTING
CC1	MUSIC_15A	JAZZ/BLUES/POP MUSIC
CC1	MUSIC_15B	JAZZ/BLUES/POP MUSIC
Syllabus: http://www.econ.berkeley.edu/.../Economics...pdf		

Table 3. A sample course, ECON2, from the articulation report. The catalog description of ECON2 beings with, "Economics 2 provides an introduction to both microeconomics, the study of consumer and firm behavior, markets, international trade, and market failures; and macroeconomics, the study of economic growth, unemployment, and inflation. Students learn both economic theory and some of the empirical evidence behind the theory. Special emphasis is placed on the application of economic tools to contemporary economic problems and policies."

Campus	*Course ID*	*Course Title*
UC1	ENGIN26	3D Modeling for Design
CC1	ENGIN_77	PROGRAMMING/MATLAB
CC1	ENGIN_22	ENGINEERING GRAPHICS
CC1	MATH_3E	LINEAR ALGEBRA
CC1	ENGIN_45	PROPERTIES/MATERIALS
CC1	MATH_3F	DIFFERENTIAL EQUATIONS
CC1	PHYS_4B	GEN PHYSICS W/CALCULUS
CC1	MATH_3C	CALCULUS III
CC1	ENGIN_35	ENGIN MECH-STATICS
CC1	PHYS_4A	GEN PHYSICS W/CALCULUS
CC1	MATH_11	DISCRETE MATHEMATICS
CC1	ENGIN_17	INTRO ELECT ENGIN
CC1	ENGIN_18	INTRO ELECTRICAL ENGIN
CC1	MATH_3B	CALCULUS II
CC1	CIS_61	STRUC/INTER COMP PRG
CC1	PHYS_4C	GEN PHYSICS W/CALCULUS
Syllabus: https://tbp.berkeley.edu/syllabi/758/download/		

Table 4. A sample course, ENGIN26, from the articulation report. The catalog description of ENGIN26 is, "Three-dimensional modeling for engineering design. This course will emphasize the use of CAD on computer workstations as a major graphical analysis and design tool. Students develop design skills, and practice applying these skills. A group design project is required. Hands-on creativity, teamwork, and effective communication are emphasized.

best model in median, mean, and std. rank metrics and was therefore used to produce the articulation report.

The primary barrier to adoption of this methodology is sharing of enrollment data. It is a challenge to successfully approach community colleges with requests to share anonymized enrollment data. In order for this endeavor to be successful, it may take the support of existing centralized repositories of these data, such as the assist.org system, operated by the UC Office of the President (UCOP), or national data collectors such as the National Student Clearinghouse or Department of Ed. A secondary barrier to adoption is the degree to which this data-assistive method is accepted into the socio-technical system of course articulation. If the method is seen as a threat to articulation officers' jobs, as AI is increasingly seen as to many jobs, it will be difficult to integrate with the articulation officer as the point of contact.

While our study focused on these methods used to identify new articulations, out of date articulations may be just as important to identify. Transfer students receiving credit for courses that do not well enough prepare them for the material that will be encountered upon transfer are also harmful to student success. The methods described could just as well be used to identify the existing articulations with the lowest articulation scores, for re-consideration.

LIMITATIONS AND FUTURE WORK

A limiting factor to the potential success of a report generated for UC1 to CC1 articulation is that CC1 is what is known as a "common feeder school" to UC1, meaning that it is a top source of transfer students for UC1. This means that articulations between the two institutions may be near saturation levels. A limitation of the course2vec approach is that a course must have an enrollment history in order to receive an embedding. This would rule out courses which are being offered for the first time as candidates for articulation to or from. In this case, a content-based model would need to be defaulted to. The primary limitation of the machine translation method is that it relies on existing articulations to learn the translation, ruling out the method for application to institution pairs for which no articulations exist, which are by definition the most in need of articulation. Again, the content-based methods could be applied in these cases and promising unsupervised language translation methods [17] may become candidates for overcoming a lack of existing articulations to train on. Faculty currently consider factors such as the difficulty of the source course and its syllabus when deciding to accept a proposed articulation. Future enhancements to the content-based models could include parsed syllabus information and data from the LMS. Were it available, test questions and their grading or even graded student answers might further provide a means for automatically scoring the match in difficulty between courses at different institutions. These data are available in MOOC datasets, and thus the emerging articulation context of MOOC micro-credentials[3] and their mapping to accredited degree programs is already halfway to a tenable context for this approach.

[3]https://www.edx.org/micromasters

RIGHTSLINK

ACKNOWLEDGEMENTS

The authors would like to thank Tammeil Gilkerson (Laney College) and Walter Wong (UC Berkeley) for contributing anonymized enrollment data to this effort. We would also like to thank Patti Ahuna, Johanna Metzgar, and Merryl Owen for their contribution of institutional knowledge on the logistics of student transfer into the UC system.

REFERENCES

1. Thomas Bailey, Shanna S. Jaggars, and Davis Jenkins. 2015. Redesigning America's community colleges: A clearer path to student success. *Cambridge, MA: Harvard University Press.* (2015).
2. California Intersegmental Articulation Council. 2013. Handbook of California Articulation Policies and Procedures. (2013). https://www.csusb.edu/sites/csusb/files/CIAC_Handbook_Spring_2013.pdf
3. United States. Government Accountability Office (GAO). 2017. Higher education: students need more information to help reduce challenges in transferring college credits. (2017). https://www.gao.gov/assets/690/686530.pdf
4. Don Hossler, Doug Shapiro, Afet Dundar, Mary Ziskin, Jin Chen, Desiree Zerquera, and Vasti Torres. 2012. Transfer and Mobility: A National View of Pre-Degree Student Movement in Postsecondary Institutions. Signature Report 2. *National Student Clearinghouse* (2012).
5. Davis Jenkins, Amy E. Brown, John Fink, Hana Lahr, and Takeshi Yanagiura. 2018. Building Guided Pathways to Community College Student Success: Promising Practices and Early Evidence From Tennessee. *Community College Research Center (CCRC)* (2018).
6. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient Estimation of Word Representations in Vector Space. *CoRR* abs/1301.3781 (2013). http://arxiv.org/abs/1301.3781
7. Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013b. Exploiting Similarities among Languages for Machine Translation. *CoRR* abs/1309.4168 (2013). http://arxiv.org/abs/1309.4168
8. Tomas Mikolov, Wen tau Yih, and Geoffrey Zweig. 2013c. Linguistic Regularities in Continuous Space Word Representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.
9. David B Monaghan and Paul Attewell. 2015. The community college route to the bachelor's degree. *Educational Evaluation and Policy Analysis* 37, 1 (2015), 70–91.
10. 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 Modeling and User-Adapted Interaction* (2019), 1–39. https://doi.org/10.1007/s11257-019-09218-7
11. 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
12. Alexandria Walton Radford, Lutz Berkner, Sara C Wheeless, and Bryan Shepherd. 2010. Persistence and Attainment of 2003-04 Beginning Postsecondary Students: After 6 Years. First Look. NCES 2011-151. *National Center for Education Statistics* (2010).
13. Jennifer B. Schanker and Erica L. Orians. 2018. Guided Pathways: the Scale of Adoption in Michigan. *Michigan Community College Association (MCCA)* (2018). Retrieved from http://www.mcca.org/uploads/ckeditor/files/SOA%20Publication%20Final(1).pdf.
14. Doug Shapiro, Afet Dundar, Faye Huie, Phoebe Khasiala Wakhungu, Xin Yuan, Angel Nathan, and Youngsik Hwang. 2017. Tracking Transfer: Measures of Effectiveness in Helping Community College Students to Complete Bachelor's Degrees.(Signature Report No. 13). *National Student Clearinghouse* (2017).
15. Jack E Smith. 1982. Articulation and the chief instructional ocfficer. *New Directions for Community Colleges* 1982, 39 (1982), 41–49.
16. Laurens Van Der Maaten. 2014. Accelerating t-SNE using tree-based algorithms. *The Journal of Machine Learning Research* 15, 1 (2014), 3221–3245.
17. Ruochen Xu, Yiming Yang, Naoki Otani, and Yuexin Wu. 2018. Unsupervised Cross-lingual Transfer of Word Embedding Spaces. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2465–2474.

RIGHTSLINK