Design and Deployment of a Better Course Search Tool: Inferring Latent Keywords from Enrollment Networks

Matthew Dong[1], Run Yu[2], and Zachary A. Pardos[1(✉)]

[1] University of California, Berkeley, CA 94720, USA
{mdong,zp}@berkeley.edu
[2] Wuhan University, Wuhan 430072, Hubei, China
run.yu@whu.edu.cn

Abstract. Liberal arts universities possess a vast catalog of courses from which students can choose. The common approach to surfacing these courses has been through traditional keyword matching information retrieval. The course catalog descriptions used to match on may, however, be overly brief and omit important topics covered in the course. Furthermore, even if the description is verbose, novice students may use search terms that do not match relevant courses, due to their catalog descriptions being written in the specialized language of a discipline outside of their own. In this work, we design and user test an approach intended to help mitigate these issues by augmenting course catalog descriptions with topic keywords inferred to be relevant to the course by analyzing the information conveyed by student co-enrollment networks. We tune a neural course embedding model based on enrollment sequences, then regress the embedding to a bag-of-words representation of course descriptions. Using this technique, we are able to infer keywords, in a system deployed for a user study, that students (N = 75) rated as more relevant than a word drawn at random from a course's description.

Keywords: Course search · Inferred keywords · Latent topics · Course2vec · Skip-gram · Higher education · Recommender systems

1 Introduction

The course catalog is often the first resource consulted by current and prospective students when wanting to familiarize themselves with the topical offerings of a university. With many universities offerings thousands of distinct courses over the span of several years, browsing through the description of each is untenable. Instead, classical information retrieval (i.e., search) using keyword matching is now offered at many, but not all, institutions. A keyword matching approach; however, is only as good as the words the description contains and the users' ability to craft a query using those words. Many course descriptions can be

M. Scheffel et al. (Eds.): EC-TEL 2019, LNCS 11722, pp. 480–494, 2019.
https://doi.org/10.1007/978-3-030-29736-7_36

overly brief, omitting topical terms from the description that are nevertheless contained in the course. Furthermore, for novice students, it can be difficult to gauge the similarity of courses in different departments because of the superficial differences in how different disciplines describe the same material.

In this paper, we seek to mitigate the shortcomings of topic omission and non-standardized keywords across disciplines in catalog descriptions by leveraging the regularizing power of machine learned embeddings. We apply neural embedding models to historic sequences of student course enrollments in order to embed courses into a space regularized by abstract features, or concepts, associated with courses. We then regress from this space to the space of course descriptions in order to add semantics to the course vectors. These semantics become the keywords which can be added to an enhanced university course search.

Showing the utility of a data mining, or technology enhanced learning approach in the real-world, sometimes called "closing the loop," is an objective of growing emphasis in the community. To integrate this modeling process into a larger design scheme that includes the deployment of this enhanced course search feature in a production level course recommender system, we first conduct a user study (N = 75) to measure the degree to which our model's inferred keywords correlate with student perceptions of relevance. Choosing six courses they have completed, students rated the relevance of keywords for each course generated from several sources, including random keyword selection baselines. Using these data, we were able to identify a probability threshold for which generated keywords were statistically significantly more relevant than words chosen randomly from the course's description. We use this threshold to dynamically determine the number of inferred keywords to display per course in the deployed search feature. The overall structure of the paper follows the process we followed for designing the enhanced search, outlined in Fig. 1.

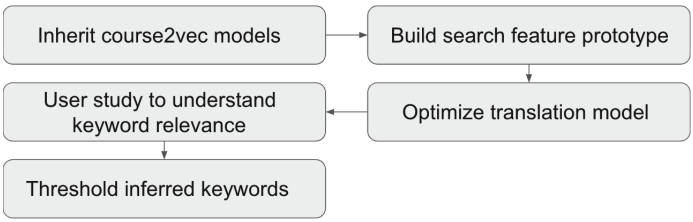

Fig. 1. Design process for the enhanced search feature

2 Related Work

Recommendation of courses and course grade prediction in formal higher education contexts has become an active area of research in data mining applied to education [1,4,15], with neural network based approaches to recommendation manifesting in deployed systems [13]. The degree of adaptivity is a significant element in deciding the type of recommendation experience a student will receive. Collaborative-based approaches, for example, have high adaptivity, whereby a student's course history is evaluated as input and suggestions are generated based on what courses the student is predicted to most likely take next. Similar approaches to social activity recommendation [5] or within-course resource recommendation have also been proposed [17]. Some shortcomings with a collaborative-based course recommendation approach are that the predicted courses may likely be courses the student already knows about, and furthermore they may be biased towards courses already popular at the university. Search is a different kind of approach, one in which a user's query represents an object (or topic) on the boundary of what the user is familiar with. In this case there is minimal adaptation, other than to the query provided. Systems taking this more knowledge-based or simple information systems approach have also seen emergence in the real-world, with one providing course evaluation and grade distributions for queried courses [3].

However, users may still experience problems in finding the information they are looking for with the classical search experience [16]. The typical approach can be improved through augmenting the search interface itself using assistive widgets [6] or by adding inferred keywords to course description, and allowing them to be matched on by the user's query. This adding of keywords to an object can be thought of as a form of classical semantic annotation [7], but with big data and modern machine learning used to generate the semantics. Mesbah et al. [8] also leverage the tagging of educational resources, such as MOOCs, using more classical natural language processing to provide the end user a synopsis of the course content. This tagging could alternatively be framed as a form of topic modeling. Motz et al. [12] provide an approach in this vein most relevant to ours in which they use students' course enrollments as a signature with which to learn themes of studying using Latent Dirichlet Allocation (LDA) [2]. Our approach is closer to the user experience of an information system but using machine learning techniques more commonly seen in collaborative-based models. We substitute LDA with the more contemporary machine regularization of skip-gram models [10] and take the work further in practical application by implementing, evaluating, and deploying it on campus.

Skip-gram neural networks are a natural choice for learning concepts, or regularities in sequential data. In the canonical example of their application to natural language, vector arithmetic, $vector[KING] - vector[MAN] + vector[WOMAN]$, results in a vector closest to $vector[QUEEN]$ [11]. In essence, the embedding has learned the concept of gender and royalty, albeit abstractly as a geometric regularity. By applying this approach to course enrollment sequences (e.g., CS101 MATH88 ECON141), we expect the skip-gram to learn similar types of concepts about courses, which we will then associate with words used to augment a course's searchable description. Prior work has found success in embedding courses in this manner, validating the model by its agreement with campus sources of course similarity [13]. We extend this application into course search and contribute a novel tuning of the semantic association process.

3 Models

Our approach to generating inferred course keywords comprises of three fundamental modeling elements: (1) a vector representation of courses learned from enrollment histories (2) a bag-of-words representation of course catalog descriptions (3) a model that translates from the enrollment-based representation to the catalog-based representation. This is essentially a machine translation, not between languages [9], but between a course representation space formed from student enrollment patterns and a semantic space constructed from instructors' descriptions of the knowledge imparted in each course.

3.1 Course2Vec

The course2vec model involves learning distributed representations of courses from students' enrollment records throughout semesters by using a notion of a enrollment sequence as a "sentence" and courses within the sequence as "words", borrowing terminology from the linguistic domain. For each student s, a 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 a 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 by maximizing the objective function of context prediction over all the students' course enrollment sequences.

It is important to stress that our method of producing a course vector from enrollments (i.e., course2vec) does **not** use any course description information. It is based only on sequences of course IDs, with no natural language used. The generalizing principal is that patterns of student collective course taking can produce representations of courses containing abstract concepts [14] of relevance to student course search. The trick to exploiting this is to associate these abstract concepts with concrete keywords, accomplished by the translation model, explained in the section after the next.

3.2 Bag-of-Words Representation

We represent course catalog descriptions using the simple but indelible approach of bag-of-words and its variants. To create a course description vector, the length of the number of unique words across all items serves as the dimension of the vector, with a non-zero value if the word in that vocabulary appears in the description. We experiment with the description vector as binary or as one of two weighting schemes described here:

- binary: value of 1 indicating that the term occurred in the document, and 0 indicating that it did not.

– tf-idf scheme [16], 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.
– custom weighting scheme such as tf-bias:

$$tf - bias = \left(\frac{number\ of\ occurences\ of\ words}{total\ word\ count}\right)^{-bias} \tag{1}$$

Empirically, lower bias has been found to produce more general words whereas higher bias produced more specific terms [14], which may be useful in surfacing course semantics at different levels of granularity.

We evaluate all three variants in our model selection phase.

3.3 Translation Model

Our premise is that there are useful concepts learned in the embedding of course2vec, but these concepts left in number form are not associated with any semantics. To associate the patterns learned in course2vec with semantics, we apply a translation from the course2vec vector to its respective natural language course description vector.

We use a multinomial logistic regression to conduct this semantic mapping, where the skip-gram based course vectors are used as input and the corresponding descriptions of every course as bag-of-word encodings are the multi-hot labels being predicted. After this model is trained, the probabilities of each word in the vocabulary belonging to a skip-gram course vector can be computed by consulting the softmax probability distribution over the entire vocabulary. Using this probability distribution, it is now possible to find the high probability words predicted based on course2vec which are NOT in the course description. These words can subsequently serve as inferred keywords in our enhanced course search.

Logistic regression is used to represent translation between languages because the spaces being translated to and from are linear vector spaces (skip-grams have no non-linear activations). However, in case the relationship between spaces in the course domain is non-linear, we evaluate a single hidden layer neural network with non-linear activation as an additional candidate translation model in our optimization experiments.

4 Experimental Environments

4.1 Off-Line Dataset

Course descriptions were sourced from the official campus course catalog API and the data was pre-processed in the following steps: (1) concatenate each description with its respective title (2) remove stop words (3) remove punctuation (4) tokenize and collect unigram and bigram phrases to constitute our vocab (5) finally compile the binary value vector, tf-idf vector, and tf-bias vector representation for each course. In addition, we filter out certain types of courses including freshman seminars and special topics courses that shared identical generic departmental descriptions and titles. A total of 6,582 courses remained in our final course dataset.

The course embeddings used in this experiment were trained and optimized to a source of validation from a previous study [13]. We inherit this course embedding from that work, where a vector size of 229 was used.

While the above are data artifacts used for the experiments reported in this paper, the data are automatically refreshed at the university, and models retrained as part of the regular maintaining of the search feature in the production system, described more in the next section.

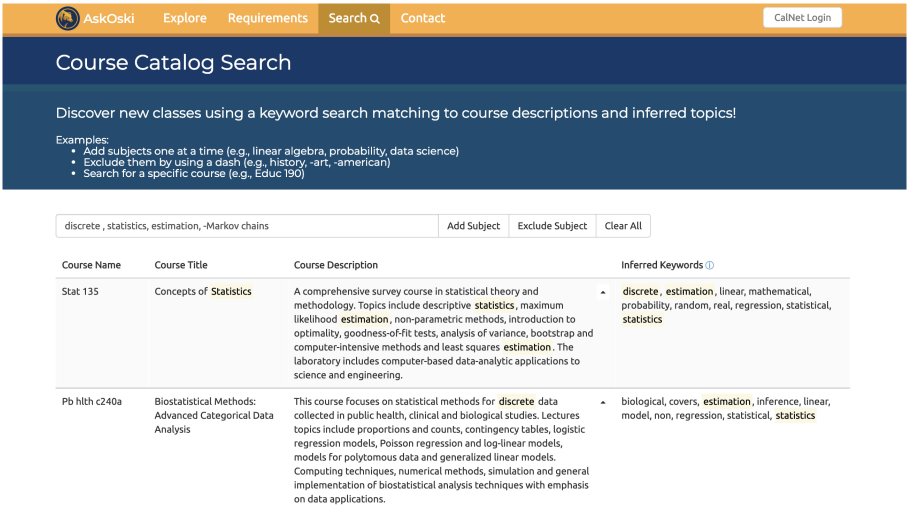

Fig. 2. A prototype of the course search feature before model tuning and user testing

4.2 Online Environment

Our first step, after inheriting a course embedding, was to apply a machine translation to the bag-of-words binary space without any optimization and design a search interface to surface the predicted words not in the course description. Figure 2 shows this prototype of the intelligent search feature as part of the campus course recommender system. Users may enter queries into the search box, which are string matched to terms in the course title, description, and inferred keywords and returns courses where any matches exist, prioritizing results that match to multiple fields. The inferred keywords serve as an additional source of semantics to match on that is intended to improve the relevancy and accessibility of the returned results. As seen in Fig. 2, the courses returned from the queries are based on keywords that do not necessarily belong to the course description, but are still relevant to the user through the inferred keywords. The keywords in this demo were produced by a model trained under default settings and validated by inspection. We simply select the top 10 predictions from the model to display in the "inferred keywords" column. This prototype exists on a beta testing server. Before deploying it to the production server, we sought to first refine the translation model and perform a user study to insure that the inferred keywords were of real relevance to students at the University.

5 Offline Model Optimization

In this section, we conduct offline predictive model experiments intended to optimize for heuristics pertinent to online user relevancy ratings. The goal was to select a single model after this optimization, that would serve as the model evaluated by real-world users in the user study phase. Because there is no offline data on student's perceptions of keyword relevancy, we came up with heuristics to optimize to as substitutes.

5.1 Tuning Parameters

Using the inherited course embeddings and course description vectors, we trained multinomial regression models and neural networks to translate from embedding to descriptions.

We experimented with different NLP representations of course catalog descriptions, serving as the labels for the translation model. The course representations were already pre-optimized so we focused on searching hyperparameters for the bag-of-words representations of their respective descriptions. We sweep a range of max document-frequency (max-df) for building the collective vocabulary, which ignores terms that have a document frequency strictly higher than the given threshold, filtering out common, often generic words found across all catalog descriptions such as "student", "semester", and "course" that are not useful as search keywords. Bag-of-words vectors are also characterized using a range of tf-bias weights and also tf-idf and binary values. We explored using a multinomial logistic versus a single hidden layer neural network to serve as the translation model. Hyperparameters in the grid search included max-df, BOW representations (binary, tf-idf, tf-bias), and translation models (multinomial logistic, 1 hidden layer neural net), totalling 144 experiment runs.

5.2 Model Selection Heuristics

In order to select which model to use in our user study, we produced the following heuristic metrics for each (all ranging from 0 to 1) and then selected the model with the highest sum of all metrics. The metrics were *recall@max_length*, *precision@10*, *department frequency*, and *distribution similarity*. The rationale for their use was as follows:

Precision and Recall. Precision and recall are meant to capture the most direct evidence of relevancy of the inferred keywords to its respective course. Precision@10, where 10 is the likely number of keywords to be shown in the search interface, is the proportion of keywords in the top 10 model predictions that also appear in the course description. Recall@max_len, where max_len is the maximum length of any description (182 words), represents the proportion of keywords found in the description if the model were to predict the entire description.

Using precision and recall alone is not sufficient in our case. A high, or perfect score for either would indicate that our model has simply learned the description of a course without capturing any additional signal surfaced from behavioral patterns. To measure the generalizability of our model in uncovering hidden semantics, we utilized two other quantifiable metrics of success, department frequency and distribution similarity, described next.

Department Frequency. Department frequency is the standard measure of document frequency in text mining, replacing document with course department. The department frequency of word w_i is:

$$dept\_freq(w_i) = \frac{number\ of\ departments\ with\ w_i}{number\ of\ total\ departments} \tag{2}$$

A department frequency of 1 indicates that a particular keyword appeared across every department. For every model trained, the average department frequency was calculated across all the words predicted. This metric is intended to measure the ability of the model to identify words from related disciplines and therefore extrapolate from the original course itself. This is intended to help overcome the lack of standardization found in the language used to describe similar courses in different departments.

Distribution Similarity. Distribution similarity is the cosine similarity between the vector of keyword frequencies from the model's predictions and the vector of uniform frequencies where each entry is the total number of possible keywords to be predicted, divided by the number of unique keywords actually predicted. This metric is intended to help us select a model that offers a more equal spread of keywords and does not overly favor a limited vocabulary, which was observed to occur during early development training phases.

Since we want to maximize each one of these metrics, our single value used for model selection is the sum across all four. Simply taking the sum has the convenient property that the combined distribution looks similar when training the regression model and the neural net, but the two are distinguishable when stratifying by each of the metrics.

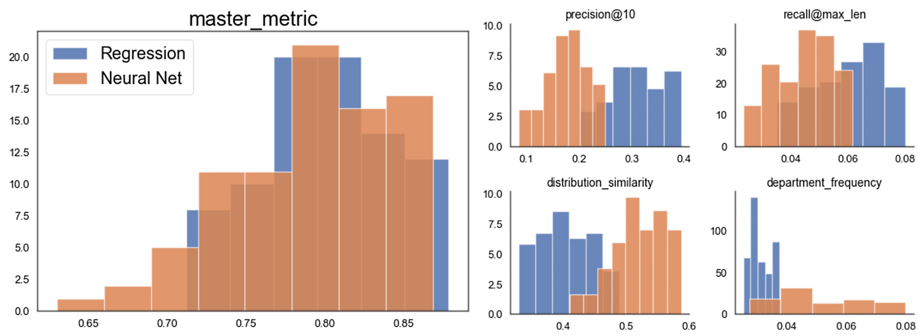

Fig. 3. Distribution of keyword evaluation metrics colored by translation model (Color figure online)

Model Evaluation. The experiment proceeds with the algorithmic optimization of our model via a grid search over the selected hyperparameters and the calculation of the described metrics for every hyperparameter set. For testing, we elected to test the model both with and without cross-validation. Because the use case of the search feature involves predicting course keywords only for existing courses rather than new courses, the model is trained on the entire dataset to allow it to learn all possible words across the collective descriptions. For thoroughness, we repeated the same grid search with 5-fold cross-validation but there was insufficient variance across each of the metrics to perform model selection.

Results of the hyperparameter search is shown in Fig. 3, where we find that a logistic regression model outperforms the neural network in terms of our relevancy heuristics (recall and precision) but the neural net outperforms the regression model by our heuristics of generalizability (department frequency and distribution similarity). We opt to use the regression model to err on the side of relevance so users are not off-put by seemingly unrelated results returned to their queries. Our optimal model and corresponding hyperparameters received the highest score sum, but was not the max precision nor recall model.

6 User Study

Following the offline experiment model selection, we follow up with a human judgment evaluation to better gauge how the model results are aligned with students' perception of relevance. A user study was conducted during which students were asked to rate keywords belonging to five different groups:

1. *Model Sorted (All)*: Top five overall keywords as predicted by the model.
2. *Model Sorted (Description)*: Top five words in the description in order of likelihood as predicted by the model.
3. *Model Sorted (Non-Description)*: Top five words not in the description in order of likelihood as predicted by the model.

4. *Random (Description)*: Five random words from within the description.
5. *Random (All)*: Five random words across all collective descriptions.

An example of these keyword groups for a select course are shown in Table 1.

Table 1. Keywords drawn from each of our five groups for STAT 135

Course: STAT 135 - Concepts of Statistics
Course Description: A comprehensive survey course in statistical theory and methodology. Topics include descriptive statistics, maximum likelihood estimation, non-parametric methods, introduction to optimality, goodness-of-fit tests, analysis of variance, bootstrap and computer-intensive methods and least squares estimation. The laboratory includes computer-based data-analytic applications to science and engineering
Model Sorted (All): regression, statistics, random, statistical, estimation
Model Sorted (Description): statistics, statistical, estimation, variance, tests
Model Sorted (Non-Description): regression, random, real, linear, discrete
Random (Description): course, engineering, includes, methods, computer-based
Random (All): diverse collection, topics problems, year credit, planning research, user interfaces

The random (all) words represent a baseline relevancy score. We expect the description groups to perform much better than this baseline and desire that the model predicted non-description words are also better than randomly selected words. The random (description) group provides the second benchmark to compare our model sorted non-description group to, quantifying how much value our enhanced search may add on top of the catalog description. These groups are not necessarily disjoint; the unique of all 5 groups were taken and randomized before showing them to the student, with an average of 18.5 unique keywords per course.

6.1 Study Design

Undergraduates were recruited from popular student Facebook groups to participate remotely in our keyword rating study in exchange for a $10 Amazon gift certificate. Study participants logged into the main *AskOski* recommender site using their University credentials in order to access the survey. The survey system looked-up the courses the student had taken and then asked them to choose six to rate the keywords of. Figure 4 shows the course selection interface for the study. Student were asked to rate solely on their experience with the class to prevent bias in keyword ratings whereby a student may be tempted to simply rate a word as relevant only if it appeared in the description.

For every keyword, students were asked for their five point Likert scale agreement with the following statement: *This keyword is relevant to the course*, where

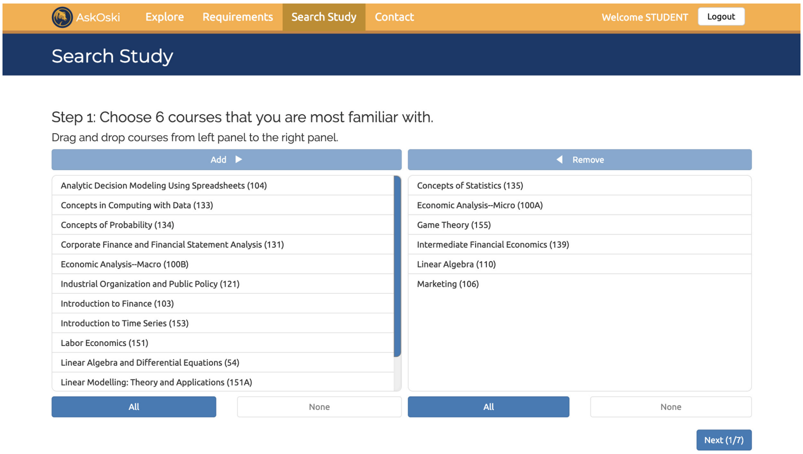

Fig. 4. Personalized survey interface after user authentication

a score of 1 corresponded with *Not Relevant At All* and a score of 5 corresponded with *Very Relevant*. A total of 75 students participated in our study, rating a total of 8,355 keywords.

6.2 Results

The average student relevancy ratings of keywords from each of the five groups is shown in Fig. 5. All three Model Sorted groups, and the Random (Description) group, scored between a 3 (neutral) and 4 (relevant) in keyword relevance. Selecting keywords at random from the entire vocabulary, Random (All), scored a 1.836 (below "Not Very Relevant"), representing students' lower bound for perception of relevance. All pairwise differences between keyword groups were statistically significantly reliable at $p < 0.05$, after applying a Bonferroni correction for multiple (10) Wilcoxon rank sum tests, except between Model Sorted (All) and Random (Description) groups, which was not statistically separable ($p = 0.019$).

The benefit of the model-based approach in terms of improving relevance of chosen keywords can be quantified by the difference in ratings between the random within-description selection group, Random (Description), 3.612, and the model-based within-description selection group, Model Sorted (Description), 3.916. A breakdown of the proportion of each rating level by group can be seen in Fig. 5. The majority (51%) of Model Sorted (Description) keywords received a 5 rating (Very Relevant), compared to Random (Description), for which 42.1% were Very Relevant. Model Sorted (Non-Description) has a much lower proportion of Very Relevant ratings (31.5%), but still considerably higher than the Random (All) baseline, with 7.3%, and with 62.3% of keywords in its group receiving the lowest relevancy rating as compared with Model Sorted (Description), that received 20.6% Not Relevant ratings.

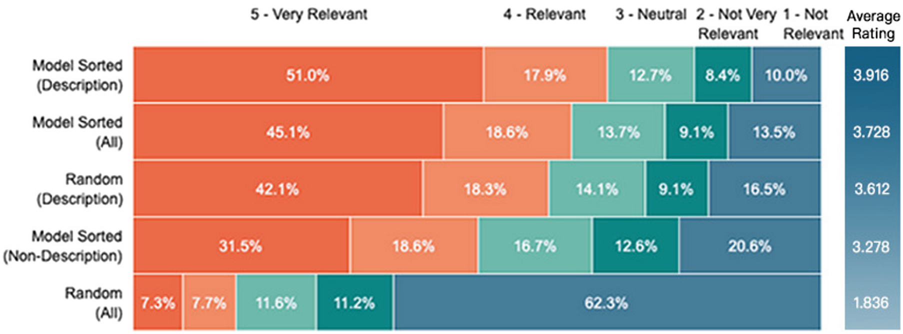

Fig. 5. User study relevancy ratings by keyword group

The way in which student relevancy ratings played out with respect to the within-group ranking of the keyword, based on model probability, is shown in Fig. 6. The average relevancy rating (y-axis) by rank (x-axis) is plotted for each of the three model-based approaches. Since the two random models do not involve any model probabilities, they also are not associated with a rank. Therefore, they are represented in the plot as horizontal lines corresponding to their averages (Fig. 5). The Model Sorted (All) trend shows the highest average ratings at rank 1, followed by an apparent asymptote down to just above the average random within-description level. Differences in ratings between these two at each rank level are statistically significantly reliable except at ranks 3 and 4. The Model Sorted (Non-Descrip) trend is initially above Random (Description) at rank 1, but then dips down and asymptotes to a Neutral average rating of 3.

A premised benefit of the predictive model was to surface relevant keywords that are not in a course's description (Non-Descrip). If we were to highlight inferred keywords, we would like to show only keywords that are "better" than words chosen randomly from the description, or at least not show words statistically significantly worse. The Model Sorted (All) ratings are statistically reliably higher than Random (Description) at ranks 1 and 2. We use this information to tailor our strategy for when and how many inferred keywords to display in the production version of our enhanced course search feature.

6.3 Selecting Inferred Keywords to Display in Search

With an improved understanding of the model predicted keywords' relevancy, we discuss how to leverage this information towards improving the search feature by updating our inferred keyword selection criteria. In the prototype, the criterion was to always display the top 10 model keywords, which did not exclude words in the description. We continue to not exclude keywords from the description, as showing them could serve the added benefit of a topic category source for reference. Thus, we choose Model Sorted (All) for this analysis.

We leverage the observation that Model Sorted ratings correlate with rank to investigate how well the underlying model probabilities of those words correlate

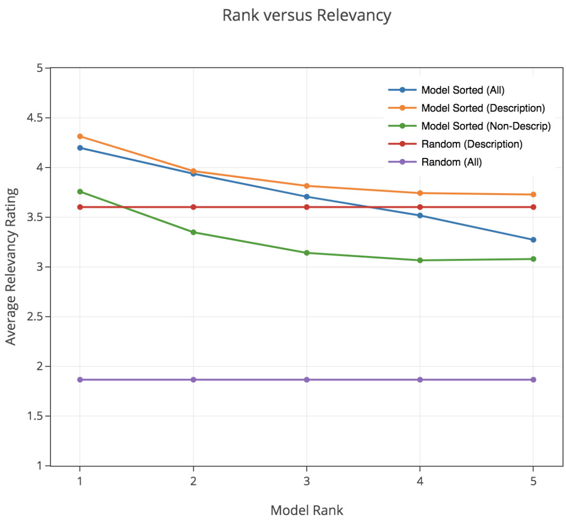

Fig. 6. Keyword group rank vs relevancy

with student relevancy ratings. If there is a correlation, then the probabilities, along with a threshold, could be used to dynamically determine which words should be included as inferred keywords on a per course basis. To conduct an analysis comparing model probabilities to user ratings, we normalize these two sets of ratings using Z-scores and then average them by Model Sorted rank. We find a substantive correlation between probability and rank and would like to choose a threshold of probability from Model Sorted (All), such that all keywords with that probability or above can generally be expected to produce keywords perceived by students to be more relevant, on average, than a word chosen at random from the description. The analysis in the previous section (Fig. 6) found that user relevancy ratings for Model Sorted (All) were significantly higher than Random (Description) at ranks 1 and 2. Therefore, we use the probability at rank 2 as the cut-off. Using this probability cut-off, we find 4.32 total words on average expected to be displayed for each course, with 2.33 within-description words and 2.00 non-description words surfaced on average within these semantics.

7 Conclusion

We explored surfacing novel, searchable semantics of a course using an embedding of courses informed by course selection histories, and supported our methodology through a user study to evaluate the relevancy of these keywords. Our experiment contributes both methodologically to the use of embeddings to surface latent semantic tags and to the design of data-driven information systems in educational settings. Our process of interface prototyping, followed by offline model optimization, user testing, and incorporation of study findings into the production software system can also serve as a design model and guide for other technologies to tune data and technology enhanced analyses towards better student learning and exploration experiences.

References

1. Backenköhler, M., Scherzinger, F., Singla, A., Wolf, V.: Data-driven approach towards a personalized curriculum. In: Proceedings of the 11th EDM Conference (2018)
2. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
3. Chaturapruek, S., Dee, T., Johari, R., Kizilcec, R., Stevens, M.: How a data-driven course planning tool affects college students' GPA: evidence from two field experiments. In: Proceedings of the 5th Learning @ Scale Conference (2018)
4. Chen, W., Lan, A.S., Cao, D., Brinton, C., Chiang, M.: Behavioral analysis at scale: learning course prerequisite structures from learner clickstreams. In: Proceedings of the 11th EDM Conference (2018)
5. Farzan, R., Brusilovsky, P.: Social navigation support in a course recommendation system. In: Wade, V.P., Ashman, H., Smyth, B. (eds.) AH 2006. LNCS, vol. 4018, pp. 91–100. Springer, Heidelberg (2006). https://doi.org/10.1007/11768012_11
6. Fessl, A., Wertner, A., Pammer-Schindler, V.: Digging for gold: motivating users to explore alternative search interfaces. In: Pammer-Schindler, V., Pérez-Sanagustín, M., Drachsler, H., Elferink, R., Scheffel, M. (eds.) EC-TEL 2018. LNCS, vol. 11082, pp. 636–639. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98572-5_62
7. Kiryakov, A., Popov, B., Ognyanoff, D., Manov, D., Kirilov, A., Goranov, M.: Semantic annotation, indexing, and retrieval. In: Fensel, D., Sycara, K., Mylopoulos, J. (eds.) ISWC 2003. LNCS, vol. 2870, pp. 484–499. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-39718-2_31
8. Mesbah, S., Chen, G., Valle Torre, M., Bozzon, A., Lofi, C., Houben, G.-J.: Concept focus: semantic meta-data for describing MOOC content. In: Pammer-Schindler, V., Pérez-Sanagustín, M., Drachsler, H., Elferink, R., Scheffel, M. (eds.) EC-TEL 2018. LNCS, vol. 11082, pp. 467–481. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98572-5_36
9. Mikolov, T., Le, Q.V., Sutskever, I.: Exploiting similarities among languages for machine translation. CoRR (2013). http://arxiv.org/abs/1309.4168
10. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
11. Mikolov, T., Yih, W.t., Zweig, G.: 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, pp. 746–751 (2013)
12. Motz, B., Busey, T., Rickert, M., Landy, D.: Finding topics in enrollment data. In: Proceedings of the 11th EDM Conference (2018)
13. Pardos, Z.A., Fan, Z., Jiang, W.: Connectionist recommendation in the wild: on the utility and scrutability of neural networks for personalized course guidance. User Model. User-Adap. Inter. **29**(2), 487–525 (2019). https://doi.org/10.1007/s11257-019-09218-7

14. Pardos, Z.A., Nam, A.J.H.: A map of knowledge. CoRR preprint, abs/1811.07974 (2018). https://arxiv.org/abs/1811.07974
15. Polyzou, A., Karypis, G.: Feature extraction for classifying students based on their academic performance. In: Proceedings of the 11th EDM Conference (2018)
16. Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. McGraw-Hill Inc., New York (1986)
17. Shani, G., Shapira, B.: Edurank: a collaborative filtering approach to personalization in e-learning. In: Proceedings of the 7th EDM Conference (2014)