

Enter a Job, Get Course Recommendations

Nan Li², Naren Suri¹, Zheng Gao¹, Tian Xia³, Katy Börner¹, Xiaozhong Liu¹

¹School of Informatics and Computing, Indiana University Bloomington

²College of Transportation Management, Dalian Maritime University

³School of Information Resource Management, Renmin University of China

Abstract

In an interdisciplinary learning environment, students are facing difficulties to locate the right education opportunities, e.g., campus courses or MOOCs, to achieve their career goals. In this paper, we propose a novel student program planning system. Using the system, students can enter job preferences, e.g., “*Software Engineer at Google*”, and text-based and network graph-based recommendation algorithms will suggest education opportunities that help students achieve their career goals. Preliminary results show that the proposed solution is promising in recommending students a personalized education plan.

Keywords: Student Program Planning; Education; Text/Graph Mining

Citation: Li, N., Suri, N., Gao, Z., Xia, T., Börner, K. & Liu, X. (2017). Enter a Job, Get Course Recommendations. In *iConference 2017 Proceedings*, Vol. 2 (pp. 118-122). <https://doi.org/10.9776/17227>

Copyright: Copyright is held by the authors.

1 Introduction

Personalized student program planning and course scheduling services have been identified as an important problem in the past few years. As students are learning in a more interdisciplinary environment, they face an increasing amount of pressure to locate the right courses and programs. In previous studies, Bendakir and Aïmeur (2006); Werghi and Kamoun (2009); Chu, Chang, and Hsia (2003) proposed a number of information recommendation and data mining methods, i.e., association rules and decision trees, in support of program planning and course recommendation by students. In most of those studies, courses were recommended to the target student based on his/her (computational) profiles.

However, students may have different career goals, and they may have different needs when choosing what courses to take. For instance, two iSchool students may take the very same required courses in the first academic year (because they come from the same academic program), but they may be interested in different courses in the second academic year because of different career goals.

In order to address this problem, we proposed a novel student programme planning system that takes current expertise and career goal information into consideration. Unlike prior efforts, we propose a new method to recommend education opportunities that meets student’s career goals, e.g., jobs. As Figure 1 shows, the proposed system lets users (actors) submit job information. Text-based and/or network graph-based search and recommendation methods are used to recommend high quality courses or MOOCs. The system represents information on jobs, university courses and MOOCs via a text index and a network graph index.

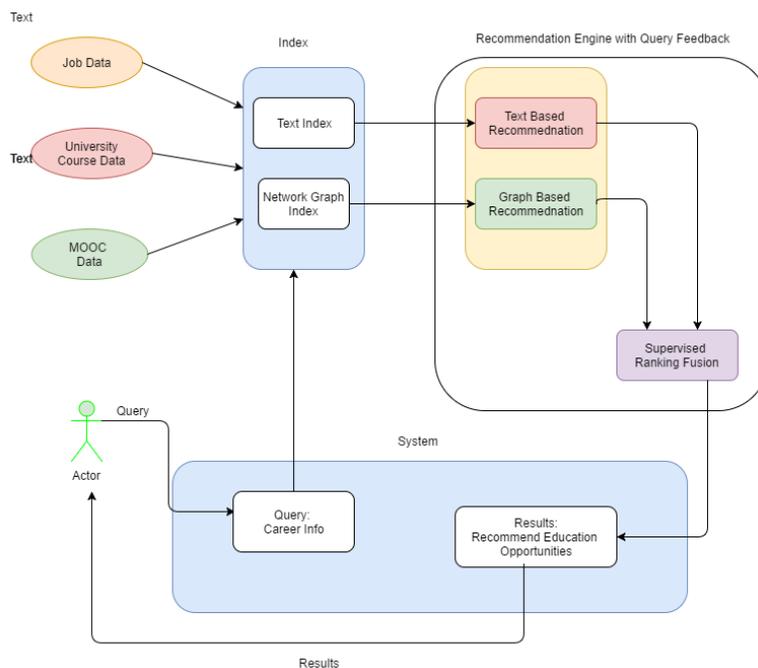


Figure 1: Course Recommender System Architecture.

3 Methodology

In this section, we describe the proposed methodology in detail along with the preliminary experiment results.

3.1 Data Collection and Indexing

For this project, we collected various kinds of data, including university course data, MOOC data, and job posting data. Using information extraction algorithms, we extracted different named entities from the text data, i.e., course/MOOC description and job positing content. See legend in Figure 2 for sample node and edge types. A listing of all different nodes types and the number of exemplars for each type can be found in the top-right of the figure.

However, entities are isolated in the text index and relationships are not explicit. For example, from a course recommendation viewpoint, *Information Retrieval*, *Information Visualization* and *Bayesian Network* should be interconnected implicitly or explicitly. In order to address this problem, we index all the courses, MOOCs and jobs on a novel heterogeneous knowledge graph that interlinks all the jobs, courses, and MOOCs via semantically typed links collected from Wikipedia¹. For instance, by extracting Wikipedia concepts from the *Data Scientist* job postings, on average, they are linked to *Machine Learning*, *Matlab*, and *Unstructured Data* nodes with transitioning probabilities (0.072, 0.024 and 0.010 respectively). The concepts (keywords) extracted from Wikipedia are also interconnected on the graph via page incoming/outgoing links.

The complete network graph has a total 395,030 nodes and 993,526 edges. There are 8,350 jobs, 716 university courses, 750 MOOCs, 1,774 companies, 6,924 company specialities, 38 job functions, 954 locations, 375,208 related entities (keywords extracted from Wikipedia), and 316 instructors.

3.2 Text-based Approach

In the system, students can use a text query to represent their career goal, and the system recommends courses and MOOCs based on the job query together with the probability that the course matches the job query, $P(course|job_{query})$. In the proposed system, we utilized a two-step approach. First, student inputs text query is sent to the job text indexation and relevant jobs are fetched. Then, we extracted the keyword information from each retrieved (and top-ranked) job posting as job_{query} . For instance, when student input *Soft Engineer* in the system, we first retrieve a number of job postings from the job index, and then extract keyword list from those postings as the job_{query} to represent student information need. Note that, the extracted keywords associate with the weight, e.g., frequency or probability in the target job postings, which can be translated to the query vector or query language model for the next step.

Meanwhile, a pseudo relevance feedback approach is used to further enhance the recommendation performance. For instance, the most important words/entities are extracted from the top ranked course/MOOC descriptions to enhance the query quality and ranking results. More detailed pseudo relevance feedback algorithm can be found in (Yu, Cai, Wen, & Ma, 2003).

3.3 Graph-based Approach

As aforementioned, a graph with job, course, MOOC and keyword nodes is constructed for graphical recommendation. On a graph G , when we query for a job from job node to retrieve the course or MOOC node, the traversed path over the edges with probabilistic weights results in ranking the highly relevant education opportunities. The results are calculated using the random walk algorithm to recommend and rank the candidate's educational opportunities. On the graph, if we use N_j to represent the query job, and N_o for a candidate's course/MOOC, the ranking score can be represented by:

$$P(N_j \rightarrow N_o) = \sum_{I_k^{N_j \rightarrow N_o}} \prod_{N_x \in I_k^{N_j \rightarrow N_o}} P(N_{x+1}|N_x)$$

¹Wikipedia 2015 Dump

	Precision	MAP	Precision@5	Precision@10	MAP@5	MAP@10
Vector Space	0.6113	0.7589	0.6250	0.6113	0.7487	0.7589
Language Model	0.7050	0.8275	0.6750	0.7050	0.7835	0.8275
Relevance Feedback	0.6550	0.7463	0.7000	0.6550	0.8198	0.7463
Graph (J-K-C)	0.5465	0.6345	0.5165	0.5465	0.6486	0.6345
Graph (J-K-K-C)	0.5425	0.6104	0.5150	0.5425	0.5953	0.6104
Graph (J-K-J-K-C)	0.4200	0.5945	0.3750	0.2333	0.4632	0.3303

Table 1: Preliminary Result for Different Ranking Features

where $I_k^{N_j \rightarrow N_o} | F(path_u)$ is a path instance belonging to a path function $F(path_u)$. The random walk probability from N_j to N_o on this path can be calculated by $\prod_{N_x \in I_k^{N_j \rightarrow N_o}} P(N_{x+1} | N_x)$, with $P(N_{x+1} | N_x)$ being the transitioning probability between the nodes on the graph.

For this method, we propose a number of different random walk based path functions on the graph, $F(path_u)$. For instance, on the proposed graph schema, the job (N_{job}) and course/MOOC nodes ($N_{opportunity}$) can be interconnected via three important path functions, $N_{job} \rightarrow N_{keyword} \rightarrow N_{opportunity}$; $N_{job} \rightarrow N_{keyword} \rightarrow N_{keyword} \rightarrow N_{opportunity}$; $N_{job} \rightarrow N_{keyword} \rightarrow N_{job} \rightarrow N_{keyword} \rightarrow N_{opportunity}$. The first one uses the direct relations between course/MOOC and job via entity information, and the second and third ones use the relationship between entities (e.g., *Information Retrieval* and *Machine Learning* are interconnected on the Wikipedia graph) and the relationship between job and entities.

3.4 Informal Evaluation

Two graduate students were asked to use the course recommendation system prototype. Each of them entered 10 text queries (e.g., job titles) and rated each recommended course/MOOC as ‘useful’, ‘just OK,’ or ‘not useful’. In Table 1, we report the performance of different recommendation functions (overall ranking performance and top recommended education opportunities accuracy). Precision, MAP (Mean Average Precision), Precision@5, Precision@10, MAP@5 and MAP@10 are reported as the evaluation metrics. In the experiment, we examine three different graphical ranking functions, $N_{job} \rightarrow N_{keyword} \rightarrow N_{opportunity}$ (J-K-C); $N_{job} \rightarrow N_{keyword} \rightarrow N_{keyword} \rightarrow N_{opportunity}$ (J-K-K-C); and $N_{job} \rightarrow N_{keyword} \rightarrow N_{job} \rightarrow N_{keyword} \rightarrow N_{opportunity}$ (J-K-J-K-C).

We find that the proposed method, both text and graph ranking functions, can be useful for student program planning with career information. Meanwhile, when using each individual ranking feature, the text-based approach outperforms the network graph ones.

4 Conclusion

In this study, we introduced a novel method and prototype system that recommends courses based on job queries. Unlike prior efforts, student can enter their career goal, and the text and graph-based recommendation algorithms can recommended optimized education opportunities, MOOCs and local courses, to the user. Even though we find the text ranking features outperform graph ones in the preliminary result, the graph recommendation features can be significant in the next ranking fusion stage (as Figure 1 shows). For instance, based on the learning to rank studies, studies (Liu, Yu, Guo, & Sun, 2014; Liu, Xia, Yu, Guo, & Sun, 2016) showed that graph-based approaches can provide more distinct ranking information, which can significantly enhance the recommendation performance (e.g., from learning to rank perspective, language model plus PageRank can outperform language model plus vector space).

In the future, we plan to integrate different ranking features (supervised ranking fusion) to further enhance the recommendation performance. In addition, we will run a formal user study to identify task accuracy and performance by different user groups that might be interested to use the system, e.g., university vs. MOOC students; full-time vs. part-time students.

References

- Bendakir, N., & Aïmeur, E. (2006). Using association rules for course recommendation. In *Proceedings of the AAAI Workshop on Educational Data Mining* (Vol. 3).
- Chu, K., Chang, M., & Hsia, Y. (2003). Designing a course recommendation system on web based on the students' course selection records. In *World conference on educational multimedia, hypermedia and telecommunications* (Vol. 2003, pp. 14–21).
- Farzan, R., & Brusilovsky, P. (2006). Social navigation support in a course recommendation system. In *International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems* (pp. 91–100).
- Liu, X., Xia, T., Yu, Y., Guo, C., & Sun, Y. (2016). Cross social media recommendation. In *Tenth International AAAI Conference on Web and Social Media*.
- Liu, X., Yu, Y., Guo, C., & Sun, Y. (2014). Meta-path-based ranking with pseudo relevance feedback on heterogeneous graph for citation recommendation. In *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management* (pp. 121–130).
- Pappano, L. (2012). The year of the mooc. *The New York Times*, 2(12), 2012.
- Parameswaran, A., Venetis, P., & Garcia-Molina, H. (2011). Recommendation systems with complex constraints: A course recommendation perspective. *ACM Transactions on Information Systems (TOIS)*, 29(4), 20.
- Werghe, N., & Kamoun, F. K. (2009). A decision-tree-based system for student academic advising and planning in information systems programmes. *International Journal of Business Information Systems*, 5(1), 1–18.
- Yu, S., Cai, D., Wen, J.-R., & Ma, W.-Y. (2003). Improving pseudo-relevance feedback in web information retrieval using web page segmentation. In *Proceedings of the 12th international conference on world wide web* (pp. 11–18).