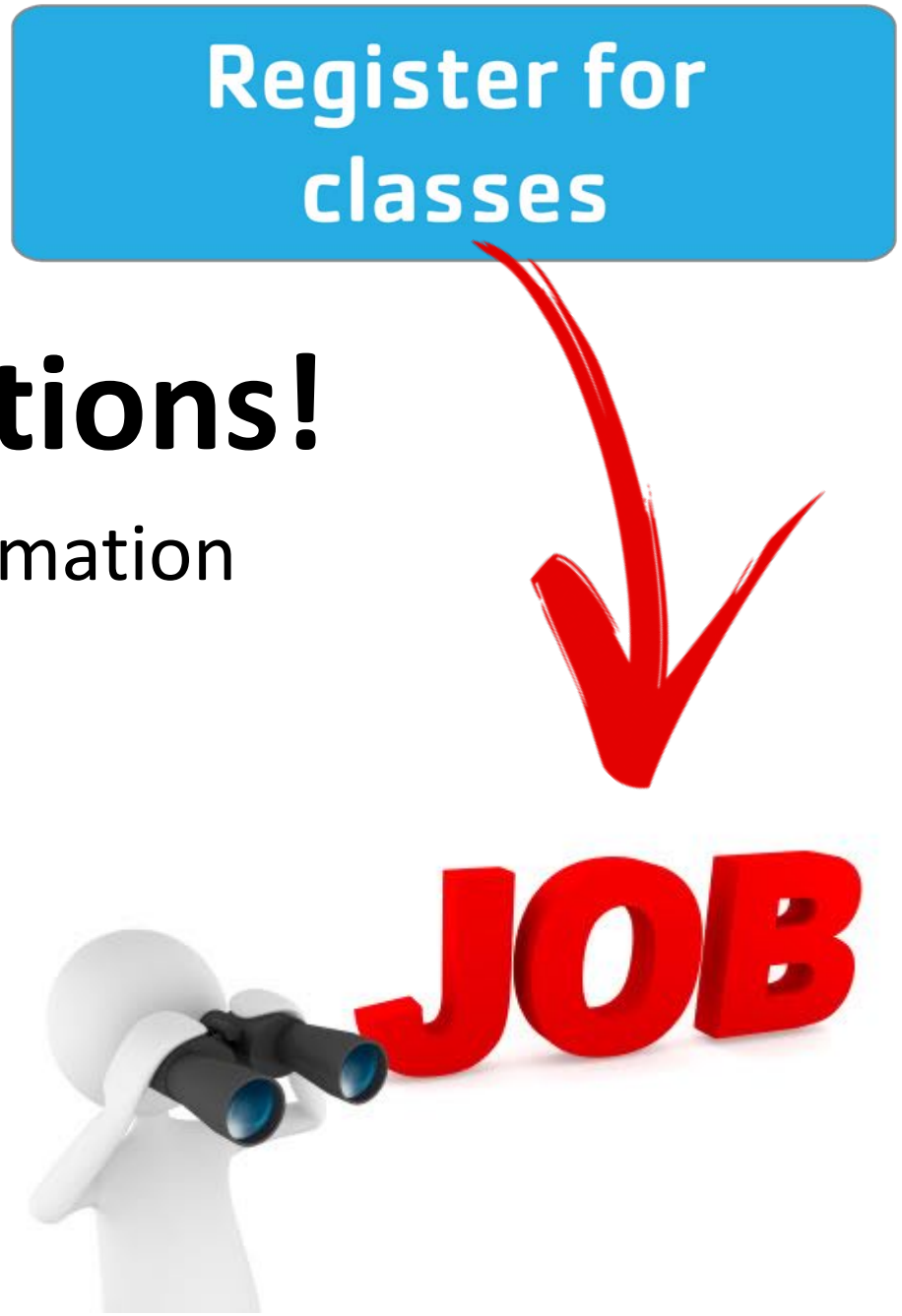
INDIANA UNIVERSITY

Enter a Job, Get Course Recommendations!

Student Program Planning with Career Information

Nan Li
Naren Suri
Zheng Gao
Tian Xia
Katy Börner
Xiaozhong Liu

iConference, Wuhan, China, March 24, 2017



Outline

Background

Methodology

- Data Collection and Indexing
- Text-based Approach
- Graph-based Approach
- Informal Evaluation

Conclusion

Research Background

- Recommend educational paths that provide a bridge from student's current status to dream job.

Personalized student program planning

Enger a job, get courses recommendations

Two iSchool students may take the very same required courses in the first academic year, but they may benefit from different courses in the second academic year because of different career goals.

Research Background

A novel student program planning system

we propose a new method to recommend education opportunities that meets student's career goals, e.g., jobs. The proposed system lets users submit job information. Text-based and/or network graph-based search and recommendation methods are used to recommend high quality university or free MOOC courses.

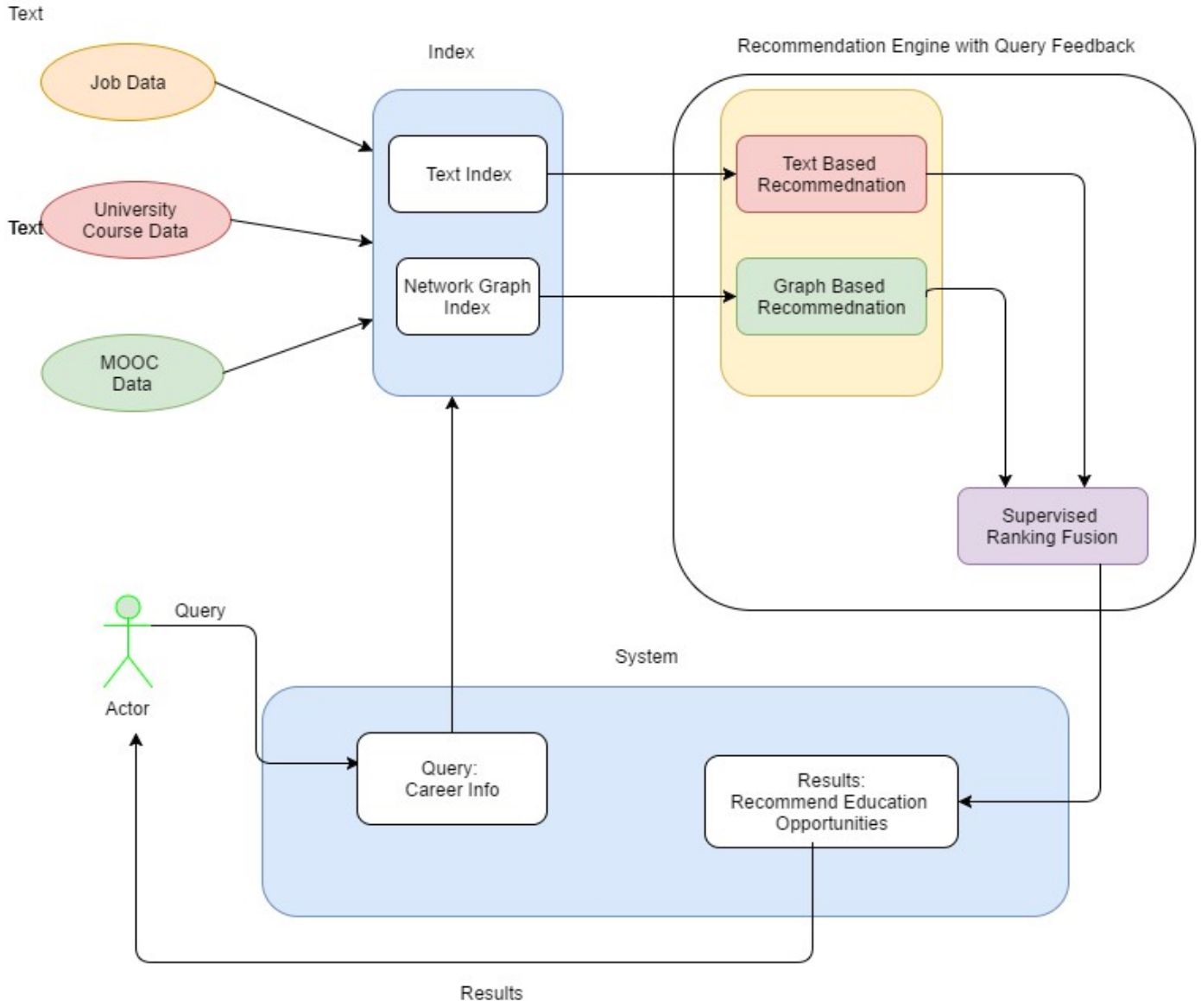


Figure: Course Recommender System Architecture

Research Background

A novel student program planning system

The system represents information on jobs, university courses and MOOCs via a text index and a network graph index. In the heterogeneous knowledge graph index, each job, company, course, topic, etc. is interconnected with nodes of other types.

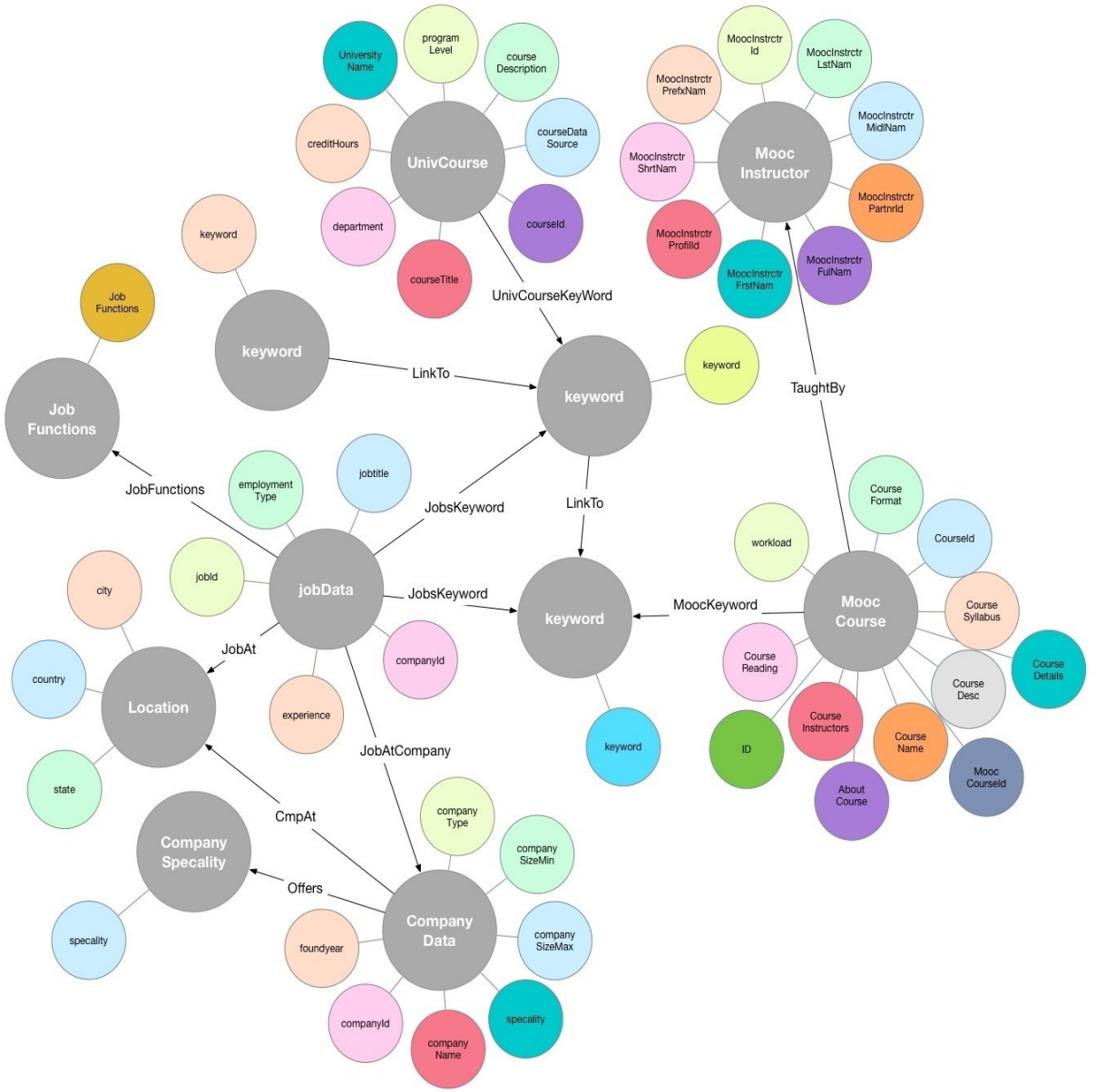
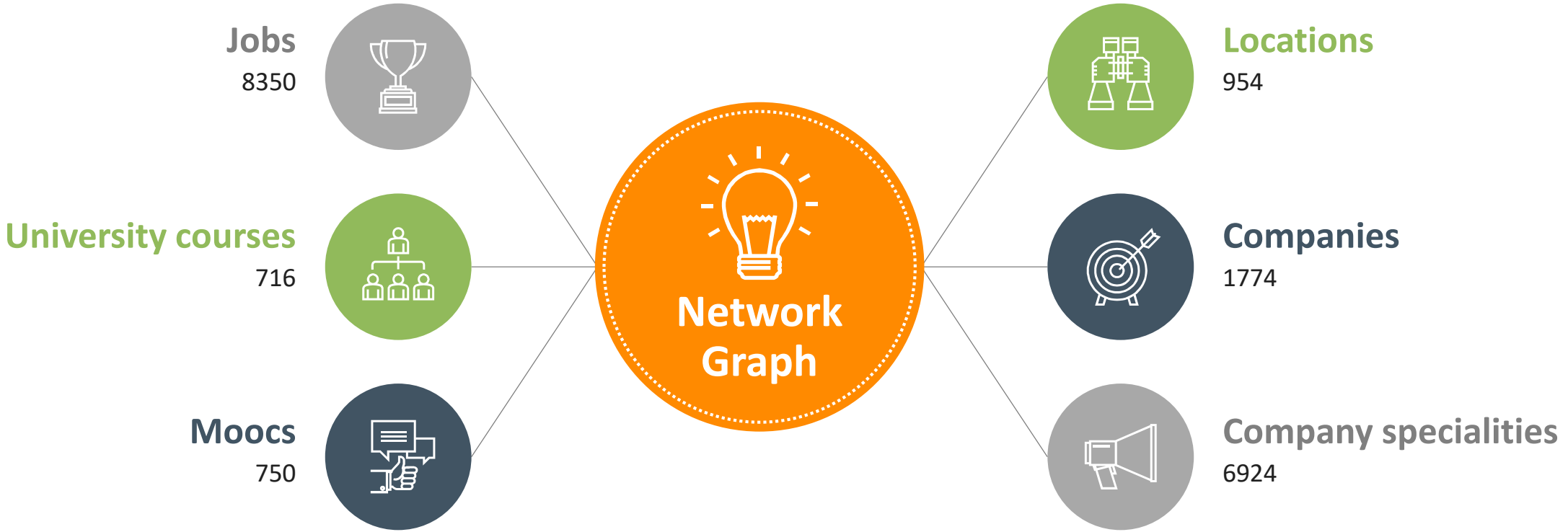


Figure: Heterogeneous Graph Index Schema Showing Different Node and Edge Types

Data Collection and Indexing

- The complete network graph has a total 395,030 nodes and 993,526 edges.



Text-Based Approach

Job query: $P(course|job_{query})$

- The query is sent to the job text indexation and relevant jobs are fetched. Then, keyword information from each retrieved (and top-ranked) job posting is extracted as job_{query}.
- Keywords associated with the weight (frequency or probability) in target job postings are extracted.

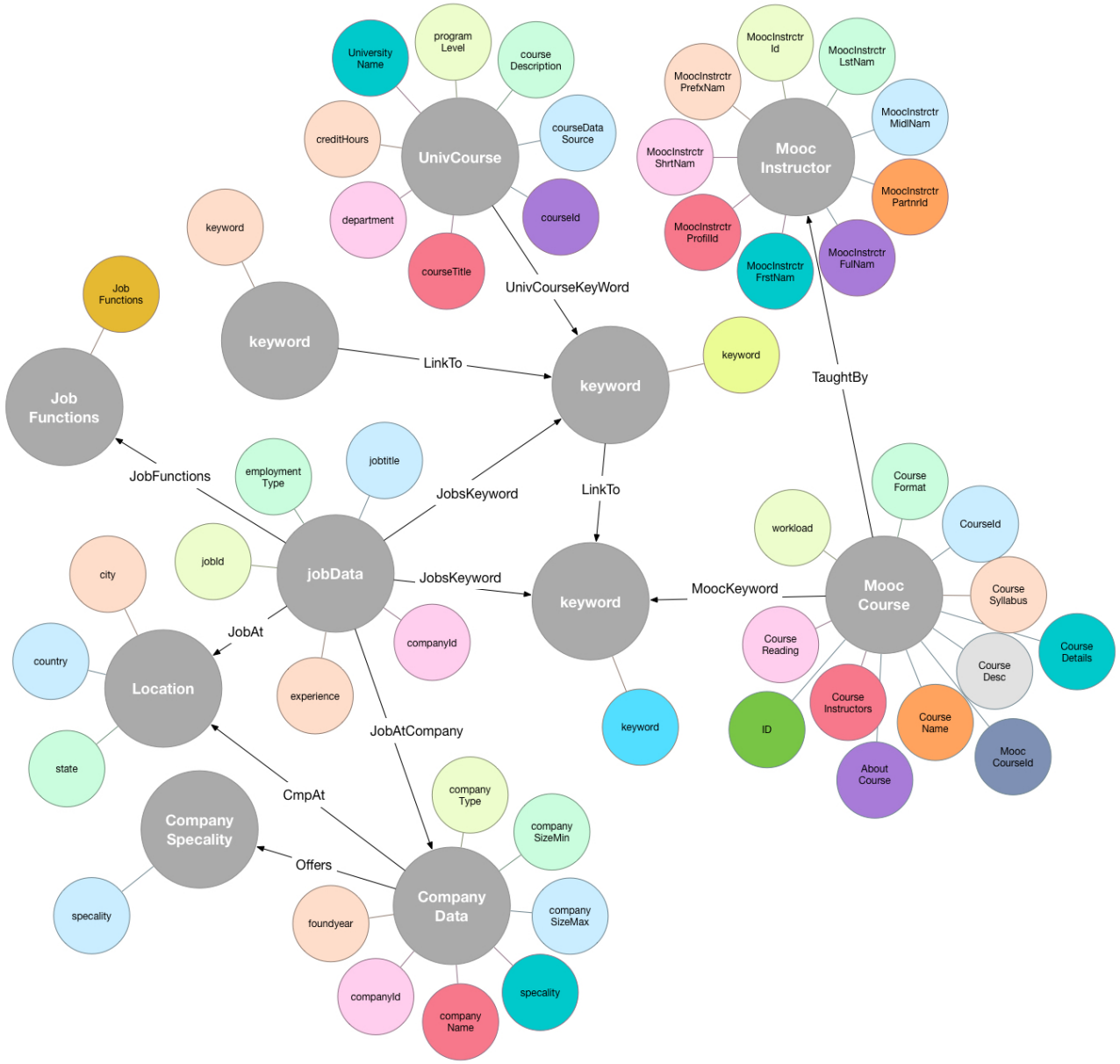
Pseudo relevance feedback approach is used to further enhance the recommendation.

Graph-Based Approach

A graph with job, course, MOOC and keyword nodes is constructed for graph-based recommendation.

Random walk algorithm is applied to recommend and rank courses:

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



Graph-Based Approach

The job and course/MOOC nodes are interconnected via three important path functions:

$$N_{job} \rightarrow N_{keyword} \rightarrow N_{opportunity}$$

$$N_{job} \rightarrow N_{keyword} \rightarrow N_{keyword} \rightarrow N_{opportunity}$$

$$N_{job} \rightarrow N_{keyword} \rightarrow N_{job} \rightarrow N_{keyword} \rightarrow N_{opportunity}$$

Informal Evaluation

The performance of different recommendation functions

	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*

- Precision, MAP, Precision@5, Precision@10, MAP@5 and MAP@10 as the evaluation metrics

Conclusions

We introduced a method and prototype system

- Students enter a career goal, i.e., dream job. The text and graph-based recommendation algorithms recommend optimized education opportunities, i.e., courses.
- Text ranking features outperform graph-bases approaches in the preliminary results. However, graph recommendation is significant in the ranking fusion stage.

Next steps

- Integrate different ranking features (supervised ranking fusion) to further enhance the recommendation performance.
- Run formal user studies to identify task accuracy and performance by different user groups.

Thank you!