# Courducate -- An MOOC Search and Recommendation System

**Qian Cheng, Yunlong Gao**

**University of Illinois, Urbana-Champaign**

**(qcheng4@illinois.edu, ygao12@illinois.edu)**

## Abstract

In this project, we build a massive open online course (MOOC) search engine, which collects over 3800 online course websites covering 5 major online course providers. We call it "*Courducate*" since it is in accordance with our vision – help people choose online *Cour*se more wisely and better e*ducate* ourselves. Our system realized two important functionalities: multi-site search and multi-filed search. In particular, the multi-field search that we presented extracts 12 important features from all the sites we crawled. Besides using the common BM25 ranking function, we also used our own ranking function to rank the sites upon a query. The evaluation shows that our system's performance is comparable with the state-of-art RedHoop course search engine [1].

## Introduction

Learning is a life-long activity that people are embracing. Along with the evaluation of Internet, Massive open online course (MOOC) has recently been experiencing a booming growth in nowadays, which brings a novel experience of learning to people. The year 2012 was even tagged as "The Year of the MOOC" by The New York Times. Our project concentrates on developing a personalized MOOC search engine that helps people search and find courses that better match with their need and capability.

Despite the fact that large amount of MOOC online courses are springing up, they are hosted by several big websites, like Udacity, Coursera, EdX, NovoED, etc., where all the course information is stored, or displayed in different format, which bring much inconvenience to the users since they cannot provide a unified platform where users can find all the information about a course that they care about. For instance, if a user is a working class, and only have time when after work, and she/he can only spend less than 3 hours a week to study a course of 'computer networks'. How could she/he do if there is no such uniformed platform? Well, in that case, the user's only choice is to search through all the websites using the keyword, and look though each and every one of them to see if the time schedule matches. Considering the general search engine, like Google, Bing, which are not specifically designed search engine for MOOC, who requires dealing with a comparatively small, but not-so-well-organized set of data.

With the previous-mentioned idea in mind, we aim to build a personalized search engine for MOOC can provide is more need-oriented service that can better match the users' requirements, such as timetable, prior knowledge, academic preferences, instructor preferences, certificate-providence, etc., which is not offered by current general search engine.

Our system realized two important functionalities: multi-site search and multi-filed search. In particular, the multi-field search that we presented extracts 12 important features from all the sites we crawled. For instance, we extracted the time information for each course, so that the course-takes can get to know about the course time schedule at the first glimpse of the course episode after getting the search result based on the query. Another example would be the university information. If the user have a higher preference of courses provided by University of Illinois over other courses, she/he could choose restrict the search result to the courses only provided by University of Illinois in the check box. Besides using the common BM25 ranking function, we also used our own ranking function to rank the sites upon a query. The evaluation shows that our system's performance is comparable with the mature RedHoop course search engine [1]. Our system provides adequate search accuracy and a user-friendly user interface. It is also able to keep the search cookies of the user so that it can facilitate search process in this way.

## Related Work

Inspired by our assignment 3, we build up our system. We also borrowed some of the idea of Redhoop [1], which is a mature online course website search engine. The advantage of the redhoop lie in two facets: i.e., across web site search and multi-field search. Redhoop covers almost every popular online course websites and it enables 8 different field search to filter out some unwanted results of the query.

Our system assembles the RedHoop in a sense that we also enables multi-site search and multi-field search. We collected over 3800 online courses, ranging from Coursera[3], Udacity[4], Edx[5], Udemy, and Khan Academy [6]. It also able to extract 12 different and important features from the crawled data, that online course takers are really caring about. Our system not only extracted feature that Redhoop did, but also added our new features: the university and instructor features. These two features are quite import for the users who have a strong academic preference over certain institutes/universities or individual instructors.

## Problem Definition

In this section, we describe the problem/challenge that we solved in more detail and in a rigorous way. We formally define the computation problem involved in your project.

We aim to design and build an online-course search system which can gather data from multiple major online course websites and provide import filtering functionality to the course-takers. The system should be providing adequate search accuracy and a user-friendly user interface. It should also be able to keep the search cookies of the user so that it can facilitate search process in this way.

The system should be able to the query from the user entered from a web interface. Moreover, the user is also asked to indicate his/her preferences over certain field of the searching by clicking on the check box listed on the left handside of the searching interface. The output of the system is the search result, ranked by the ranking function. The result should include necessary information about a course, such as the title, the instructor, the course description, etc.

The challenges of this project lie in the following aspects:

1) The formats of different online course websites are different. Thus extracting the above-mentioned features should be performed under site-specific inspection. Also extracting the user-concerned features are important to the utility of the system.

2) Implementing the ranking of the system so that the results are more relevant to the user query is quite crucial.

3) Design and implement a user-friendly web interface so that user feels comfortable when doing the search work.

## Methods

In this section we present in detail how we solved our problem and conducted our study. We discussed our method in the following subsections more thoroughly.

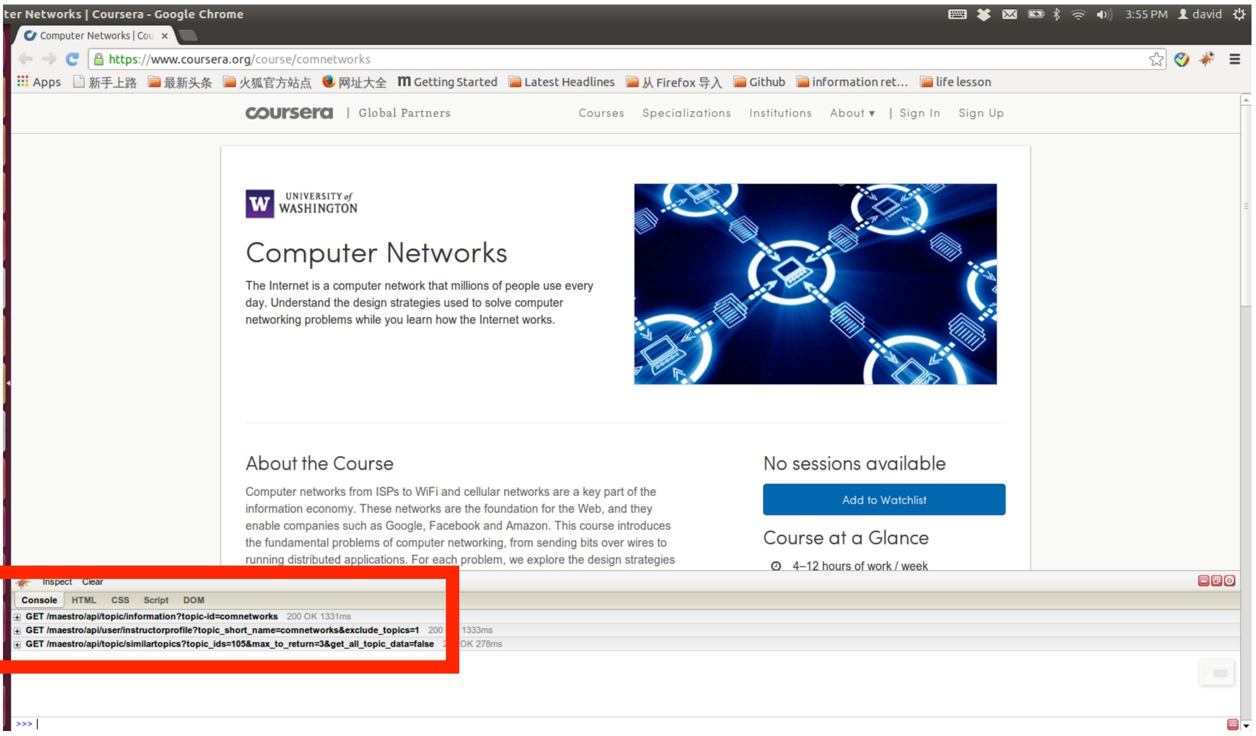
### *1. Data Crawling*

Since our system should be able to implementing the functionality of multi-site search, the first and furthermost task of ours would be crawling course data from the major course websites. We focus on five major open online course websites: Udemy [6], Udacity[4], Coursera[3], Khan Academy[7], and Edx[5]. The reason why we build our system based on their data set is not only because their categories of courses range from liberal arts to modern technology, but also their course provider are from world-famous institutions and universities, which provide great diversity to our dataset.

The challenge of crawling mainly comes from Coursera[3], whose webpage are dynamically generated and presented. Even though we used a popular crawling toolkit phantomjs[8], the necessary information of the course is still not gathered successfully. Then we looked deeper into the course website via the

| | udemy<br><br>(outerHTML) | udacity<br><br>(outerHTML) | khan | coursera (special website) | edx (innerHTML) |
|---|---|---|---|---|---|
| | | | | | |

inspection tool, as shown in Figure 1.



Computer Networks

The Internet is a computer network that millions of people use every day. Understand the design strategies used to solve computer networking problems while you learn how the Internet works.

About the Course

Computer networks from ISPs to WiFi and cellular networks are a key part of the information economy. These networks are the foundation for the Web, and they enable companies such as Google, Facebook and Amazon. This course introduces the fundamental problems of computer networking, from sending bits over wires to running distributed applications. For each problem, we explore the design strategies

No sessions available

Course at a Glance

4–12 hours of work / week

```
GET /maestro/api/topic/information?topic-id=comnetworks   200 OK 1331ms
GET /maestro/api/user/instructorprofile?topic_short_name=comnetworks&exclude_topics=1   200 OK 1333ms
GET /maestro/api/topic/similartopics?topic_ids=105&max_to_return=3&get_all_topic_data=false   2__ OK 278ms
>>> |
```

**Figure 1.** Sample MOOC website from Coursera.

We found out that the *real url* that each courser course gets are shown as in the inspection window circled by red square. We replaced the urls of coursera with the url listed in the inspection window and finally got the information we wanted.

One more thing to be noted is that we crawl both the pure text of a site and the html file of a site. And the rationale behind it is that we want not only the text description of a site but also the formatted information of a course site, which is convenient for the next step.

## *2. Data Parsing*

In this section, we presented the data format we extracted from the five sites we cared about. In particular, we wrote a python script to parse and match the keywords in each file we crawled in order to extract those important information. The detail of parsing field information for each website is shown in Table 1.

| | | | | | |
|---|---|---|---|---|---|
| price | (meta property="udemy_com:price" content=) | <span class="actual-price"> | free | all free | all free |
| duration | What am I going to get from this course?</label> | <span class="interval-length"> | NA | html tag | (Course Length: </span>) |
| hours per week | NA | <ul class="cr-iul bulleted-ul darker-bull"> | NA | html tag (special coding?) | (Estimated effort: </span>) |
| level | for all | <!-- Course level --> | for all | for all | for all |
| certificate | (Certificate of Completion </li>) | (<li>Verified certificates</li>) | non | (eligible_for_certificates) | non |
| category | <metproperty="udemy_com:category" content=) | NA | NA | html tag | NA |
| provider | easy | easy | easy | easy | easy |
| university | NA | NA | NA | NA | School: </span>) |
| instructor | (<u><strong>About the Instructor:</strong></u>) | (<h4 class="h-slim">Instructor</h4>) | NA | html tag | NA |

**Table 1.** Parse the webpages and extract the features that are important to the users.

## *3. Indexing and Search Implementation*

After crawling and extracting the features from the crawled data, we used Apache lucene[9] to build up index among the files. In particular, we stored each field in Lucene, and made them indexable by using the built-in functions of Lucene. We the user query is passed by, Lucene will call its parsing function to parse the query according to each field that has been stored.

We design a novel ranking function based on current ranking functions like MB25 and Dirichlet prior. BM25 algorithm contains both TF-IDF parts, in which IDF part has 1 element:

$$log\ \frac{N - df + 0.5}{df + 0.5}$$

TF part has 2 elements, one tf part and one qtf part:

$$\frac{(k_1 + 1) * tf}{k_1 * ((1 - b) + b * \frac{|D|}{avgdl} + tf))} * \frac{(k_3 + 1) * qtf}{k_3 + qtf}$$

The IDF can be replaced by other IDF functions introduce the effect of maximum DF like:

$$log\ \frac{N - df + 0.5}{df + MaxDF}$$

The tf part represents term frequency so the term frequency can be smoothed by Dirichlet-prior as well:

$$P(w|d) = \frac{(k_1 + 1) * ((1 - \lambda) * \frac{tf}{|d|} + \lambda * P(w|C))}{k_1 * ((1 - b) + b * \frac{|D|}{avgdl} + ((1 - \lambda) * \frac{tf}{|d|} + \lambda * P(w|C)))}$$

The query is usually short so that qtf normalization can stay the same. Thus the complete ranking function is:

$P(w|d) =$

$$log\ \frac{N - df + 0.5}{df + MaxDF} * \frac{(k_1 + 1) * ((1 - \lambda) * \frac{tf}{|d|} + \lambda * P(w|C))}{k_1 * ((1 - b) + b * \frac{|D|}{avgdl} + ((1 - \lambda) * \frac{tf}{|d|} + \lambda * P(w|C)))} * \frac{(k_3 + 1) * qtf}{k_3 + qtf}$$

### *4. User Interface Design*

The front end of user interface is basically implemented with HTML, Javascript and CSS. While the server-end is implemented with PHP to dynamically passing variables and searching results between front-end and back-end. The initial search page and result page are shown in Figure 2 and Figure 3.

## Evaluation

After we built all parts of our search engine, we are able to input test queries to see if the return results efficient enough. Specifically, for MOOC search, the evaluation contains 2 hierarchical tiers: 1) evaluation with plain text queries and 2) evaluation with comprehensive queries.

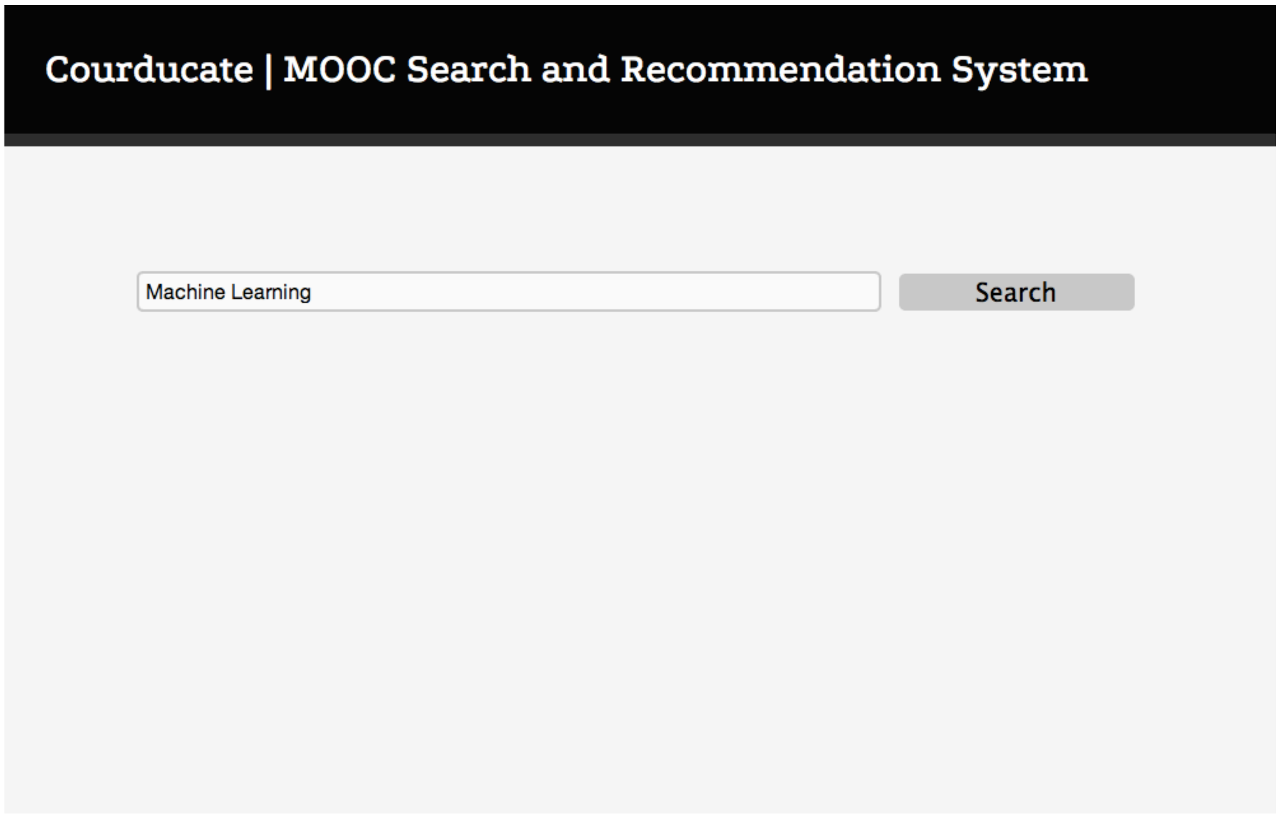


**Figure 2.** Courducate index page. Plain-text query input.

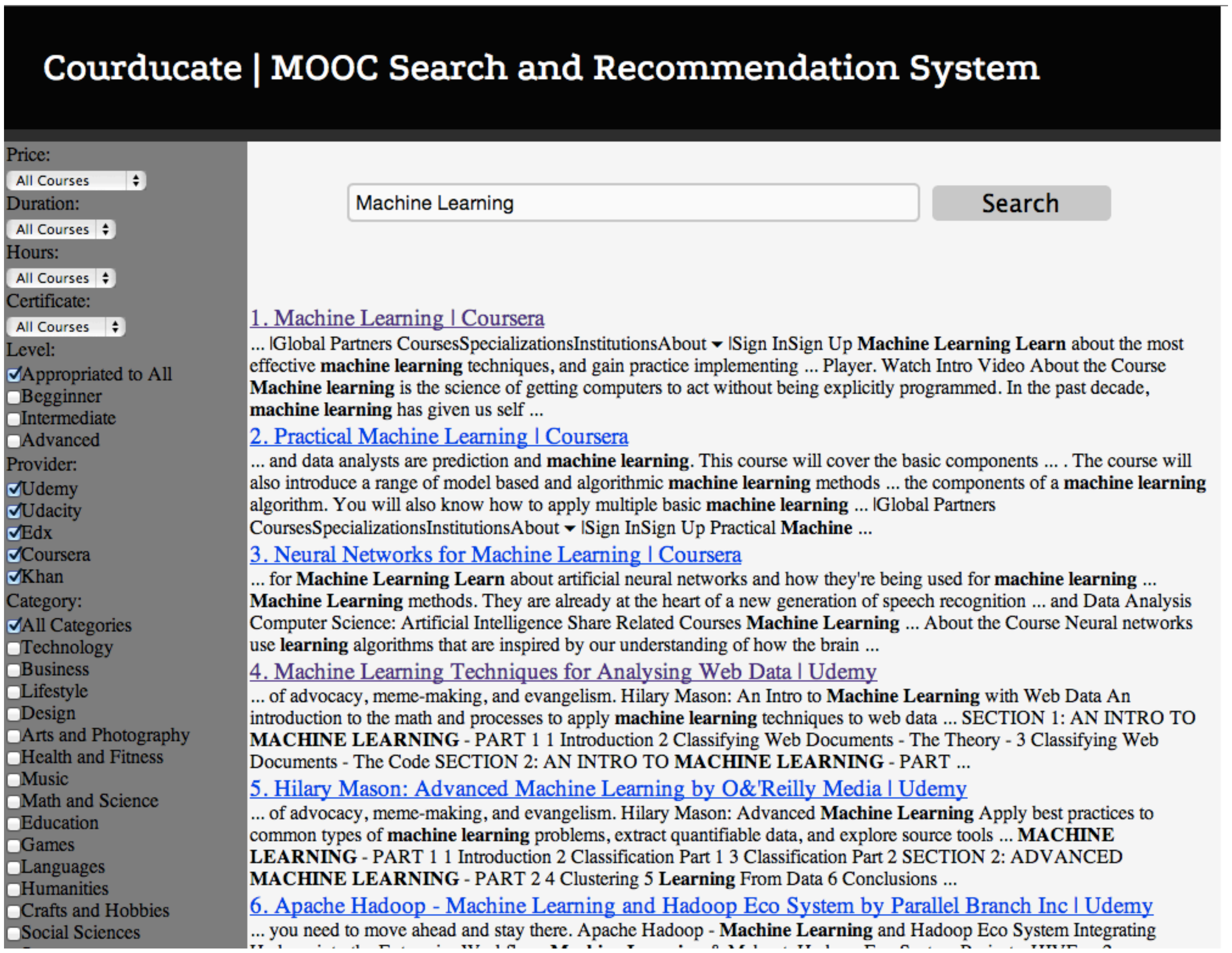
***1. Sample Results***

Before the evaluation, we will first show sample results for each type of queries. For plain text queries, we only input some course related terms like "machine learning", "Art and Design", *etc.* The web interface was shown in Figure 2. After clicking the "Search" button, Courducate will direct the user to the result page, which is also a query page supporting the second-tier search, shown in Figure 3. When clicking on the titles displayed in the result list, the user will be redirected to the course web page. For the comprehensive queries, the user can specify their course search restrictions such as:

1. Is the expected course is free or not?
2. What's the approximate duration of the course?
3. What's the recommended effort hours  for the course?
4. Does the course provide certification of completion?
5. What is the difficulty level of the course*, e.g.* appropriate to all, beginner, intermediate or advanced.
6. Specific course providers: Udemy, Udacity, Edx, Khan Academy or Coursera.

7, Specific category or fields, *e.g*. Technology, Business, Lifestyle, *etc.*

All search queries in this page will be second-tier queries with search field restriction. The user interface will memory the previous query setting and pass it to the current result page, in order to help users to gradually explore the desired courses.



Price: All Courses
Duration: All Courses
Hours: All Courses
Certificate: All Courses
Level:
✓ Appropriated to All
Begginner
Intermediate
Advanced
Provider:
✓ Udemy
✓ Udacity
✓ Edx
✓ Coursera
✓ Khan
Category:
✓ All Categories
Technology
Business
Lifestyle
Design
Arts and Photography
Health and Fitness
Music
Math and Science
Education
Games
Languages
Humanities
Crafts and Hobbies
Social Sciences

1. Machine Learning | Coursera
... |Global Partners CoursesSpecializationsInstitutionsAbout ▾ |Sign InSign Up **Machine Learning Learn** about the most effective **machine learning** techniques, and gain practice implementing ... Player. Watch Intro Video About the Course **Machine learning** is the science of getting computers to act without being explicitly programmed. In the past decade, **machine learning** has given us self ...

2. Practical Machine Learning | Coursera
... and data analysts are prediction and **machine learning**. This course will cover the basic components ... . The course will also introduce a range of model based and algorithmic **machine learning** methods ... the components of a **machine learning** algorithm. You will also know how to apply multiple basic **machine learning** ... |Global Partners CoursesSpecializationsInstitutionsAbout ▾ |Sign InSign Up Practical **Machine** ...

3. Neural Networks for Machine Learning | Coursera
... for **Machine Learning Learn** about artificial neural networks and how they're being used for **machine learning** ... **Machine Learning** methods. They are already at the heart of a new generation of speech recognition ... and Data Analysis Computer Science: Artificial Intelligence Share Related Courses **Machine Learning** ... About the Course Neural networks use **learning** algorithms that are inspired by our understanding of how the brain ...

4. Machine Learning Techniques for Analysing Web Data | Udemy
... of advocacy, meme-making, and evangelism. Hilary Mason: An Intro to **Machine Learning** with Web Data An introduction to the math and processes to apply **machine learning** techniques to web data ... SECTION 1: AN INTRO TO **MACHINE LEARNING** - PART 1 1 Introduction 2 Classifying Web Documents - The Theory - 3 Classifying Web Documents - The Code SECTION 2: AN INTRO TO **MACHINE LEARNING** - PART ...

5. Hilary Mason: Advanced Machine Learning by O&'Reilly Media | Udemy
... of advocacy, meme-making, and evangelism. Hilary Mason: Advanced **Machine Learning** Apply best practices to common types of **machine learning** problems, extract quantifiable data, and explore source tools ... **MACHINE LEARNING** - PART 1 1 Introduction 2 Classification Part 1 3 Classification Part 2 SECTION 2: ADVANCED **MACHINE LEARNING** - PART 2 4 Clustering 5 **Learning** From Data 6 Conclusions ...

6. Apache Hadoop - Machine Learning and Hadoop Eco System by Parallel Branch Inc | Udemy
... you need to move ahead and stay there. Apache Hadoop - **Machine Learning** and Hadoop Eco System Integrating

**Figure 3.** Courducate search result page. The results of the input query are displayed at the result container on the right-bottom of the page, while on the left there is a side-bar for restricted search. On the right top there is a text input box for starting a new search.

***2. Evaluation rubrics***

Due to the nature of online open courses, the set of course documents is dynamically changes. (some courses disappears while new courses become available.) In our project the system do not update the course set dynamically so that our course list is sort of out-of-date, thus with different data pool, we cannot directly evaluate our search result with those existing course search systems (e.g. Class Central and RedHoop). Thus, the strategy for evaluation of our search result is based on manually relevance judgments. We selected 5 different plain-text queries to briefly evaluate Courducate search engine:

a) Computer Science

b) Machine Learning

c) Yoga

d) Cook in home

e) iphone and ipad app development

For the comprehensive queries, we select one query above (iphone and ipad app development) and set up restrictions like: PRICE: free course; CERTIFICATION OF COMPLETION: YES; PROVIDER: Udacity & Edx. For each query we apply precision at 10 for evaluation.

***3. Plain-text query evaluation***

For 5 plain-text queries, the Precison at 10 are liseted in Table X1. We can see that for plain-text queries formed by standard noun terms reach high precision at 10. The overall precision at 10 is 0.92.

| Query | Precision@10 |
|---|---|
| Computer Science | 1.0 |
| Machine Learning | 0.7 |
| Yoga | 1.0 |
| cook in home | 0.9 |
| iphone and ipad app development | 1.0 |
| Overall | 0.92 |

***4. Comprehensive query evaluation***

For plain-text query "iphone and ipad app development", we modify the restriction settings and do search under each circumstance. The precision at 10 was listed in Table X2. The overall precision at 10 is 0.63. The evaluation of provider constraint returns a extremely low precision, which is due to lack of relevant courses provided by these 2 course websites in our document.

| Settings | Precision@10 |
|---|---|
| Free Course | 0.8 |
| Certification of Completion | 1.0 |
| PROVIDER: Udacity & Edx | 0.1 |
| Overall | 0.63 |

## Future Work

At this point, we have a basic online search engine for MOOC, which is capable to return proper results for specific course search. However, to build a personalized course recommending system, there are still several functionalities remaining to be implement.

Firstly, on lexical level, there are additional works needed to obtain precise documents and queries. For plain-text query, a spell-check tool is needed to help get the correct lexicon terms. Now if mis-spelling happens in the query, the searching result will be very inaccurate. For field information retrieval, even though we are able to locate the context describing this field, we are still not able to obtain the precise description such as the length of course, the weekly effort, *etc*. Since different course providers offer such information in different ways, an NLP analysis might be applied in prior to better retrieve more accurate field information.

Secondly, a database is required to be established to store basic user information and logs of activities. This database will be the foundation of personalized course recommendation. After creating such a database, with the stored user information, we need to explore appropriate ways to predict users' behavior so that we can provide course recommendation by their course history and preference.

## Conclusion

In this project we introduce Courducate, a search engine for massive open online courses. The search engine is based on Lucence core functions and extended to be competitive for course search. We also design a practical user interface for general users to search courses from our website. Within the provided course documentary, the search engine is evaluated to be efficient to reach desired courses from general queries and conditional settings.

## Appendix

***Individual Contributions***

Person 1: Qian Cheng: parse data, implement front end UI and back end server, presentation and report

Person 2: Yunlong Gao: crawl and analyze data, implement index and search function of the search engine, presentation and report

## References

[1]RedHoop,www.redhoop.com

[2]410 assignment 3, https://wiki.engr.illinois.edu/display/timan/Assignment+3

[3]Coursera, www.coursera.org

[4]Udacity, www.udacity.com

[5]Edx, www.edx.com

[6]Udemy, www.udemy.com

[7]Khan Academy, https://www.khanacademy.org/

[8]phantomjs, www.phantomjs.org

[9]lucene, http://lucene.apache.org/