# Best Course to Pick in NYU

Xihua Yang
Courant Institute of Mathematical Sciences
251 Mercer St, New York, NY 10012
xy644@nyu.edu

## Abstract

*Present a query system that takes a user query about a subject as input, and returns top 20 ideal courses in NYU which are relevant to the subject as output. Data is crawled from NYU Albert course search and Google search results.*

## 1. Introduction

### 1.1. Objective

The system aims at finding top 20 courses in NYU that are most relevant to a user query. The results are drawn from all NYU courses and ordered by their relevance to the query, courses that have the same titles and set up by the same institute but different course numbers are considered as different sessions of the same course, and the system only keeps the first course among them.

### 1.2. Architecture

The system is a single server runs on internal CIMS network, loads data and finds target courses. CGI module supports access from external network, reads user query, send the query to server, and display the results returned on a web page. Since the dataset is small enough to handle in memory, the system does not include a database.

### 1.3. External software and web resources

- NLTK 3.0.2[1]
- Scrapy 0.18.4[3]
- WordNet[6]
- Google Search

### 1.4. System feature

- Language: Python 2.6
- Dataset: NYU Albert course search

## 2. Experiment

### 2.1. Dataset

Before building the system, I first crawled every course available in Fall 2015 semester in NYU from NYU Albert course search, including their schools, departments, titles, short description(if applicable). Courses that have the same title, department and school information but different course number are considered as different sessions of the same course. At this stage, over $54\%$ of courses are missing short description, this would give a great impact on following process.

To solve this problem, for each course "nyu+course title" is used as query to search on Google, the first 3 results are kept as supplementary documents. If a course already have a short description, the supplementary documents is given only a low credit in ranking, otherwise a higher credit.

Another method is also considered, which is start from the website of each department, crawl down 2 or 3 more steps, and link the results to courses. However, after doing research on different courses in different institute, I found that the description of courses may appear on too many sites in too many kinds of format for crawling. Even if crawled the data is crawled, it could be difficult and inaccurate to link a document to a specific course correctly. Therefore this method is not adopted.

Due to Google's restriction on web crawlers, only ne request can be sent every 20 seconds, thus the supplementary documents are still under crawling. Meanwhile, 22187 out of 26486 courses are supplemented.

### 2.2. Modeling

Let $R_i(query, course)$, $R_t(query, course)$, $R_d(query, course)$, $R_s(query, course)$ be the normalized relevance score between a query and institute, title, description, and supplementary documents of a course respectively. Let $Q_{exact}$ be the exact query that user types in, and $Q_{synset}$ be a make up query that every word within original query is converted into a list of synonyms using WordNet in NLTK. Before passing query to score functions, every word in the

query is stemmed. Also, every word in dataset is stemmed at preprocessing stage, so that a more accurate TF-IDF[5] score can be calculated. Now relevance score $S$ between a query and a particular course can be defined as follow:

$$S = F(Q_{exact}, course) + \alpha F(Q_{synset}, course) \quad (1)$$

where

$$\begin{aligned} F(query, course) = &\beta_1 R_i(query, course) \\ &+\beta_2 R_t(query, course) \\ &+\beta_3 R_d(query, course) \\ &+\beta_4 R_s(query, course) \end{aligned} \quad (2)$$

If a course does not have a description, its supplementary documents will be taken as its description, and its supplementary documents are set to empty.

$R_i$ and $R_t$ work similarly. First calculate the total number of matches $N$ between query and corresponding course information, then multiply $N$ by $P^\lambda$, where $P$ is the percentage of words in query that have a match and $\lambda$ is a constant between 0 and 1. In this way, for query "machine learning", a title containing two "learning" will have a lower credit than the title "machine learning". When constructing these functions, I have tried to calculate their PMI value between queries and titles base on Wikipedia corpus, but it turned out to be too slow for a real time query system.

$R_d$ and $R_s$ also work similarly. They calculate the TF-IDF weighted similarity $S$ between the query and corresponding document, and produce a vector $V$, where $V_{word_k}$ is the number of matches for $word_k$. After that $S$ is multiplied by $P^\lambda$ same as above, then multiplied by mean value of $V$, and divided by standard deviation of $V$. The intuition behind this is, if a course is only relevant to "development" but not "software", it will have a higher standard deviation for $V$, which in turn will give a lower credit than a course which is both relevant to "software" and "development" when their total number of matches are close.

The parameters are tuned empirically. First set all parameters to a fix float number 1.0, and fix the query to "software development". After that look at the results and measure performance using rank of 8th relevant document[2], while the user decide which one is 10th relevant document since no labeled data is presented. Apply binary search to change the value of parameter $P_1$ until obtained the best performance. After parameter $P_1$ is tuned, use the same method to tune another parameter $P_2$ until all parameters are tuned. In final version, $\alpha = 0.2, \beta_1 = 0.2, \beta_2 = 0.75, \beta_3 = 1.2, \beta_4 = 0.35$.

### 2.3. Result

For testing, he following queries are used: "computer science", "machine learning", "software development", "art design". Words such as "I", "wish", "of", "to" are considered as stop words[4] and will not be a part of the query, therefore the queries tested are formed by seperate keywords instead of daily sentence.

The reason why choosing these 4 queries are: "computer science" is a baseline, which should contain mostly courses in Computer Science department, departments such as Data Science, Mathematics are also acceptable; "machine learning" is to check if this system can distinguish course that are truly relevant to "machine learning" from those are only relevant to "machine" or "learning"; "software development" is to check if this system can find courses such as "software engineering" even when there are much more courses related to "development" than courses related to "engineering"; at last "art design" is to see if the system generalizes well rather than performs well only in Computer Science related fields. An advantage of using these 4 queries for testing is there are enough relevant courses in NYU for measuring performance.

Results for "computer science"(ranked from 1 to 20):

```
1.Introduction to Computer Science
2.Selected Topics in Computer Science
3.Directed Study Directed Study in
   Computer Science
4.Introduction to Computer Science
5.999X PHD DISSERTATION IN COMPUTER
   SCIENCE
6.ADVANCED PROJECT IN COMPUTER SCIENCE
7.Advanced Algorithms Theory of
   Computation
8.Introduction to Web Design and
   Computer Principles
9.Computer Science Capstone Seminar
10.Pre college Computer Science
11.Introduction to Computer Programming
12.Fundamentals of Computer Science
13.Teaching of Secondary Computer
   Science
14.GUIDED STUDIES IN COMPUTER SCIENCE
15.SELECTED TOPICS IN COMPUTER SCIENCE
16.Special Topics in Computer Science
17.Introduction to Computer Programming
18.CS UY 200 Intro to Computer Science
19.Foundations of Computer Science
20.Mathematical Techniques for Cs
   Applications
```

All courses are related to "computer science". By looking at detail information, only "Teaching of Secondary Computer Science" is not set up by Computer Science department, but it is set up by Mathematics Education department, which is acceptable.

Results for "machine learning"(ranked from 1 to 20):

1.Introduction to Machine Learning
2.Machine Learning and Data Mining
3.Foundations of Machine Learning
4.Machine Learning and Computational Statistics
5.Machine Learning
6.Literature and Machines
7.Literature and Machines
8.Deep Learning
9.Minds and Machines
10.Learning In an Out of School
11.Seminar in Experiential Learning
12.Inquiries Into Teaching Learning I
13.Men and Machines
14.Doctoral Sem in Teaching Learn Culture Lang
15.Minds and Machines
16.Humans Machines and Aesthetics
17.Machine Vision
18.Learning to Speak First Second Lang Acquisition
19.Drawing Machines
20.Inquiries Into Teaching Learning III

Top 5 courses are all closely related to "machine learning", another course "Deep Learning" is at 8th place.

Results for "software development"(ranked from 1 to 20):

1.Programming in Python and Fundamentals of Software Development
2.Software Engineering I
3.FINANCIAL SOFTWARE ENGINEERING LABORATORY
4.Transportation Systems and Software
5.Software Engineering
6.Software Engineering
7.FUNDMNTLS SOFTWARE DVLPMT
8.Application Architecture Design Development
9.ADAPTIVE CONTROL SIMULATION AND SOFTWARE
10.Systems Development Analysis
11.Community and Civic Engagement in Real Estate Development
12.Software Engineering II
13.The Virtual Producer Software Instruments and FX
14.Political Environment of Development
15.Sociology of Development in Global Perspective
16.Gender Development The Political Economy
17.Software Engineering
18.Software Engineering
19.Regulation of Re Develop
20.Affordable Housing Finance and Development

Comparing to previous tests, this test is not as satisfying. However 6 out of top 8 courses are still closely related to "software development". Three more "Software Engineering" courses are ranked below 10th place, the reason could be lack of description and inaccurate supplement documents, but they are ranked among top 20 eventually.

Results for "art design"(ranked from 1 to 20):

1.Art Technology Design
2.Scenic Design in the Performing Arts Theatre Dance Film and Television
3.Design and Ornament in Islamic Art
4.The Arts Fundamentals of Painting Design
5.Collaborative Art Fundamentals of Stage Design and Production
6.Thinking Art
7.Global Histories of Art
8.Media History Art Design and Technology
9.DESIGN OF HVAC SYSTEMS
10.Urban Arts Workshop
11.Urban Arts Workshop New York
12.VISAR AD 166 Foundations of Art History I
13.Good Design Scale
14.Broadcast Design II
15.Advanced Media SeminarPrinciples of Art and Design
16.Architectural Design Art Installation in Florence
17.Capstone Design Project l
18.Site Specific Performance Art Activism Public Space
19.Planning Process in The Visual Arts
20.Capstone Design Project II

Althought courses about "art design" is not very familiar to Computer Science students, most courses are obviously related to "art design" closely, which indicated a well generalized system.

### 2.4. Analysis

In general, the system performs well for most queries, top 5 courses are mostly accurate, but top 20 may contain some courses that are not really related to the query and needs to be examine by user. The reason could be lacking of labeled data. If labeled dataset is given, a more powerful model can be trained using machine learning techniques.

For further improvement, labeling a subset of data and applying unsupervised training method may obtain better parameter set; viewing the description and supplement documents as an ordered sequence instead of simple word bag to preserve the information about co-occurence. Besides, if more storage space is given, saving Wikipedia corpus locally to calculate PMI value between queries and course titles may also help increasing accuracy.

## References


[1] S. Bird, E. Klein, and E. Loper. *Natural Language Processing with Python*. O'Reilly Media, 2009.

[2] E. Davis. Web search engines: Lecture 4: Near duplicate pages / evaluation, 2015. http://www.cs.nyu.edu/courses/spring15/CSCI-GA.2580-001/lec4.html.

[3] P. Hoffman. Scrapy, 2008. http://scrapy.org.

[4] MySQL. Stopwords. http://www.ranks.nl/stopwords.

[5] K. Sprck Jones. A statitstical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972.

[6] P. University. About wordnet, 2010. http://wordnet.princeton.edu.