Graduate School Information Retrieval Engine

Hyun Joon Cho (hcho34)

Sanghyun Choi (schoi60)

Abstract

Unlike applications to colleges where each school's expectation from students are more or less standardized, such as high SAT scores, strong GPA, excellent recommendation letters, and involvement in numerous extracurricular activities, applications to graduate schools require detailed information about each school's specific academic needs. By doing research for each school, an applicant evaluates his or her "fit" to a program offered by a particular school. In many cases, this is a very time consuming process and applicants struggle to find the right sources of information. Therefore, what we propose in this project is a 'Graduate School IR Engine' to solve such problem. It is an interactive query system that provides comprehensive information about top-ranked schools for different fields, faculty members and their research interests, current graduate students, past admission results, and prediction of receiving an admission based on machine learning approach – all in a single package.

1. Introduction

Potential applicants to graduate schools often find that they have no clue where to begin the application process. There are two main reasons behind such difficulty. First is that there is no single point of entry into doing research about schools. Applicants have to manually surf around the web to gather vast amount of information which is scattered all over the place. Second reason is that it is hard for applicants to gauge their level of academic maturity against the level sought by schools of their interest.

A typical research process might involve consulting the U.S. News to get a sense of which schools are prominent in the field of interest and, once the list of schools are narrowed down, visiting each school's homepage for faculty information. In addition, applicants may seek information about currently enrolled students to view their achievements at the time of admission and, by visiting popular websites such as GradCafe (and GoHackers.com for Korean applicants), query past admission results for extra information.

In this project, we seek to resolve the two main problems described above by simplifying the lengthy research process. The IR engine we propose has the potential to be more robust by incorporating other popular websites that we are not yet aware of such as admission results sharing site in China or India. However, we limit the websites to the most popular ones in this project as a proof-of-concept. As a distinctive feature, we provide a machine learning based predictor (binary-classifier) for determining whether an applicant is suitable for the program.

2. Features

The Graduate School IR Engine is a Python based script which is largely divided into five main features. These particular features were built on the assumption described in the previous section regarding a typical school research process. By running the main script (i.e. GradInfo.py), users can interactively query the information they need as shown in figure 1. Subsequent sections explain each individual feature in more detail.

```
============================================================
==    Welcome to the 600.466 Graduate School IR Engine
============================================================

OPTIONS:
  1 = View top schools for specified area
  2 = View faculty members for specified school(s)
  3 = View current students for specified school(s)
  4 = View admission history for specified school
  5 = Predict chance of admission for specified school
  6 = Quit

============================================================
```

Figure 1: Entry point for Graduate School IR Engine

2.1. Top Schools List

This feature is a simple crawler specifically designed for U.S. News website's graduate school ranking information [3]. Applicant may use this feature to get an overview of which schools are highly ranked in their intended field of study. By doing so, they may be able to narrow down the list of schools they are interested in and use the rest of the features to dive deeper into these schools. As shown in figure 2, users can select a broad range of fields to view the results.

```
Choose area:
1 = Business
2 = Education
3 = Engineering
4 = Law
5 = Medicine
6 = Nursing

3

http://grad-schools.usnews.rankingsandreviews.com/best-graduate-schools/top-engineering-schools/eng-rankings
engineering :
1 Massachusetts Institute of Technology | Cambridge, MA | $46,400 per year (full-time)
2 Stanford University | Stanford, CA | $48,720 per year (full-time)
3 University of California-Berkeley | Berkeley, CA | $11,220 per year (in-state, full-time); $26,322 per year
4 California Institute of Technology | Pasadena, CA | $43,710 per year (full-time)
5 Carnegie Mellon University | Pittsburgh, PA | $42,0
...
```

Figure 2: Users can choose the area of study and view the rankings

2.2. Faculty Members List

In order to scrape the list of faculty members for a school (and department) specified by the user, we heavily rely on Google's search API and fully trust that their first search result, when given the search string '{school name}+{department}+faculty', would contain the link to the

school's official faculty website. Once our crawler access this link, it assumes that the first page it sees is the list of faculty members and further assumes that it will contain anchors that have links to detailed information about a specific faculty member. So the crawler accesses each link provided the anchors and checks whether the new page contains anchors with texts like 'publications', 'teachings', 'curriculum vitae', etc. and headings like 'research interest' and 'area of interest'. When such criteria are met, the crawler looks for texts containing the following keywords:

research, holds, received, area, director, member, fellow, earned

Since many faculty members website follow the structure described above, the crawler obtains surprisingly accurate results. For example, once we query for Johns Hopkin's computer science department, we obtain the following results:

1. Yanif Ahmad
Ahmad studies and designs novel abstractions for large-scale data management. He is affiliated with the Data Management Systems Lab; Computer Systems Research Group and the Institute for Data-Intensive Engineering and Science. His research spans foundations and applications, with the K3 project realizing programming abstractions for declarative, democratized construction of distributed data systems, and the Molecular Dynamics Database pursuing data-intensive computing architectures and analytics for large biological datasets. Ahmad received his Ph.D. from Brown University in 2009.

…

35. David Yarowsky
Yarowsky's research focuses on word sense disambiguation, minimally supervised induction algorithms in NLP, and multilingual natural language processing. He earned his bachelor's ('87) in computer science at Harvard University and his master's ('93) and Ph.D. ('96) in computer and information science at the University of Pennsylvania.

…

2.3. Current Students List

Unlike the list of faculty members for a school (and department), the list of current students for a school (and department) is rarely shown or visible in a single webpage. Although some schools have the list of student names, a few personal websites, and contact information, the majority of schools do not have such information. So instead we have come up with a different approach to gather student information. Since we think that the user of our program, potential applicants to graduate schools, will be more interested in knowing what current graduate students of inquired school and department have done in the past and are doing currently rather than just a name, this feature provides a summary of the current students.

This feature uses Indeed [4], an online job search and post platform, to gather publicly posted individual data. When a user looks for students in a particular university and department, our program runs an internal query, {'title': 'PhD candidate' + 'school': 'school name' + 'fieldofstudy': 'department'}, on Indeed. Because of the limited number of resumes in a specific department of university, in the case where there is no information retrieved, the same

query without the ‘fieldofstudy’ value will be run, and its result will be displayed to the user.

For example, once we query for Johns Hopkin School of Medicine, we obtain the following results:

```
======================================================================
Name : Laura Gottschalk
----------------------------------------------------------------------
[Work Experience 1]
Company Name     : Johns Hopkins University
Company Location : Baltimore, MD
Position         : PhD Candidate
Dates            : April 2010 to Present (6 years)
[Work Experience 2]
Company Name     : Georgia Institute of Technology
Company Location :
Position         : Research Technician II
Dates            : May 2006 to May 2009 (3 years)
[Work Experience 3]
Company Name     : University of Georgia
Company Location :
Position         : Undergraduate/Graduate Assistant
Dates            : November 2003 to May 2006 (2 years, 6 months)
[Work Experience 4]
Company Name     : Department of Plant Pathology University of Georgia
Company Location :
Position         : Undergraduate Assistant
Dates            : May 2004 to May 2005 (1 year)
[Work Experience 5]
Company Name     : University of Georgia
Company Location :
Position         : Undergraduate Assistant
Dates            : January 2002 to May 2002 (4 months)
----------------------------------------------------------------------
```

Figure 3: Users can view current graduate students’ background information

2.4. Past Admission Results

We examine two popular websites, that we know of, where applicants share their admission results: GradCafe [1], a U.S website, and GoHackers [2], a Korean website. Scraping GradCafe is simple as it provides a table of structured set of data based on query parameters embedded in the URL. On the other hand, GoHackers raises some challenges as finding tests scores, GPA, and other background information has to be relied on regular expression matches. Moreover, each individual post has to be accessed via separate links so the crawler spends a significant amount of time jumping to different posts, which is resolved to a certain extent by multi-threading these accesses.

For example, once we query for Johns Hopkin’s PhD program in biology, we obtain the following results:

GradCafe Results:

Total : 17 posts

	GPA	GRE	Program (Season)	Decision	St1	Date Added	Notes
0	3.97	167/170/4.50	Biochemistry, Cellular And Molecular Biology (...	Rejected	A	3 Apr 2016	Attended one of the interview weekends and rea...
1	3.00	160/155/4.50	Biochemistry Cell And Molecular Biology (BCMB)...	Rejected	A	24 Jan 2016	
2	3.00	160/155/4.50	General Biology, PhD (F16)	Rejected	A	17 Jan 2016	
3	3.16	168/162/5.00	Biochemistry And Molecular Biology, PhD (F15)	Rejected	A	2 Feb 2015	Logged into website to check status and found ...
4	3.05	169/170/3.50	The Biochemistry, Cellular And Molecular Biolo...	Rejected	U	27 Jan 2015	
5	4.00	164/162/4.00	Bioinformatics And Computational Biology, PhD ...	Rejected	I	14 Feb 2014	GPA above from grad studies. Generic email. Ma...
6	3.10	163/169/6.00	Chemical Biology, PhD (F13)	Accepted	A	8 Mar 2013	Interview last weekend, very happy
7	3.51	163/161/5.00	Biochemistry, Cellular And Molecular Biology (...	Accepted	A	28 Feb 2013	
8	3.60	590/800/3.00	Human Genetics And Molecular Biology, PhD (F13)	Rejected	A	5 Feb 2013	
9	3.68	153/165/5.00	Chemical Biology, PhD (F13)	Accepted	A	23 Jan 2013	4 semesters + 16 x 40hour weeks of research
10	3.90	700/700/0.00	Cell, Molecular, And Developmental Biology (CM...	Rejected	A	28 Feb 2012	Rejected
11	3.84	154/170/3.50	Chemistry-Biology Interface Program(CBI), PhD ...	Rejected	I	2 Feb 2012	
12	3.92	690/780/5.50	Biochemistry, Cellular And Molecular Biology (...	Rejected	I	30 Jan 2012	Email with name . Guess lack of research exper...
13	3.54	550/690/4.50	Biochemistry, Cellular And Molecular Biology G...	Rejected	A	7 Feb 2011	Interviewed with Pharmacology & Molecular Scie...
14	3.60	800/720/3.50	Biochemistry, Cellular And Molecular Biology (...	Rejected	I	8 Mar 2010	most frustrated...
15	3.40	690/720/5.00	Cell, Molecular, Developmental Biology And Bio...	Accepted	A	5 Mar 2010	I emailed one of the directors with a question...
16	3.40	690/720/5.00	Cellular, Molecular, Developmental Biology And...	Accepted	A	26 Feb 2010	yay :o)

GoHackers Results:

Total : 2 posts

	GPA	GRE	Decision	St1	Research	WorkExp
0	3.94	590/800/4.0	Accepted	I	True	True
1	3.96	660/800/3.0	Accepted	I	True	True

Figure 4: Users can view admission results

2.5. Admission Predictor

This is an experimental feature where we make use of machine learning based approach to predicting application outcomes. First we create feature vectors with following components:

Floating point number between 0 and 1			Raw score	Binary values (1 if true, 0 otherwise)		
GPA	GRE Verbal Score	GRE Quant Score	GRE Writing Score	Research Experience	Work Experience	International Student

The first three components are normalized to floating point numbers between 0 and 1 to account for different formats while the last three are binary numbers (e.g. 1 if an applicant has research experience and 0 otherwise). Based on these 7-dimensional feature vectors, we then perform PCA for dimensionality reduction. Finally, we fit two machine learning models: *k*-Nearest Neighbors and Support Vector Machines with RBF kernel. Section 3 discusses the optimal choice of parameters of these models.

The biggest challenge of this feature involves dealing with the low quality of data. Many applicants usually leave out the test scores and GPA. Very few people share their research or work experience (i.e. number of publications or number of years active in industry) Moreover, applicants who share their background in detail are usually the ones who got accepted to a school. Other rejected applicants generally tend to leave comments that are not useful. Hence there is inevitable bias in the data. The best we could do is to filter out only the data that has all the information filled in, which means that there is very few training data. Due to this limitation, we restricted our experiments to schools and fields which returned relatively large amount of data.

The following result is based on Princeton's economics PhD application data (80% of data is used for training while the remaining 20% is reserved for testing).

```
Number of Training Data: 78
Number of Testing Data: 20
```

Classification using k-Nearest Neighbors
Accuracy: 0.75 (15/20)
Details:

	GPA	GRE(V)	GRE(Q)	GRE(W)	Work Exp.	Research Exp.	Status	Decision	Predicted
0	0.89	0.89	0.99	4.50	0	0	0	0	0
1	1.00	0.94	1.00	6.00	0	0	0	1	0
2	0.97	0.96	1.00	4.50	0	0	1	1	1
3	0.96	0.95	1.00	5.50	0	0	1	0	0
4	0.96	0.94	0.98	4.00	0	0	1	0	0
5	0.99	0.98	0.99	5.00	0	0	0	0	0
6	1.00	0.94	0.98	4.00	0	0	0	0	0
7	0.98	0.92	1.00	3.50	0	0	1	1	0
8	1.00	1.00	1.00	5.00	0	0	0	0	0
9	0.99	0.99	0.98	5.50	0	0	0	0	0
10	1.00	0.96	0.99	4.50	0	0	1	1	1
11	0.91	0.91	1.00	3.50	0	0	1	0	0
12	0.93	0.96	1.00	5.00	0	0	1	1	1
13	0.93	0.90	1.00	4.50	0	0	0	1	0
14	1.00	0.93	0.99	5.00	0	0	0	0	0
15	1.00	1.00	0.91	3.50	0	0	1	1	0
16	0.99	1.00	1.00	5.50	0	0	0	0	0
17	0.88	1.00	1.00	5.00	0	0	0	1	0
18	0.97	0.91	1.00	5.50	0	0	0	0	0
19	1.00	0.99	0.97	6.00	0	0	0	0	0

Classification using SVM with RBF kernel
Accuracy: 0.75 (15/20)
Details:

	GPA	GRE(V)	GRE(Q)	GRE(W)	Work Exp.	Research Exp.	Status	Decision	Predicted
0	0.89	0.89	0.99	4.50	0	0	0	0	0
1	1.00	0.94	1.00	6.00	0	0	0	1	0
2	0.97	0.96	1.00	4.50	0	0	1	1	1
3	0.96	0.95	1.00	5.50	0	0	1	0	0
4	0.96	0.94	0.98	4.00	0	0	1	0	0
5	0.99	0.98	0.99	5.00	0	0	0	0	0
6	1.00	0.94	0.98	4.00	0	0	0	0	0
7	0.98	0.92	1.00	3.50	0	0	1	1	0
8	1.00	1.00	1.00	5.00	0	0	0	0	0
9	0.99	0.99	0.98	5.50	0	0	0	0	0
10	1.00	0.96	0.99	4.50	0	0	1	1	1
11	0.91	0.91	1.00	3.50	0	0	1	0	0
12	0.93	0.96	1.00	5.00	0	0	1	1	1
13	0.93	0.90	1.00	4.50	0	0	0	1	0
14	1.00	0.93	0.99	5.00	0	0	0	0	0
15	1.00	1.00	0.91	3.50	0	0	1	1	0
16	0.99	1.00	1.00	5.50	0	0	0	0	0
17	0.88	1.00	1.00	5.00	0	0	0	1	0
18	0.97	0.91	1.00	5.50	0	0	0	0	0
19	1.00	0.99	0.97	6.00	0	0	0	0	0

3. Experiments

In order to choose the optimal parameters for PCA, *k*-NN, and SVM models, we performed experiments by varying the parameters and observing the resulting accuracy. The relevant parameters are the number of components to keep for PCA, the number of neighbors to assign for *k*-NN, and the gamma value for RBF kernel in SVM. The following is the experiment result:

Classification using k-Nearest Neighbors

	Num. Dimensions	Num. Neighbors	Accuracy
0	1	1	0.65
1	1	3	0.50
2	1	5	0.50
3	1	7	0.50
4	1	9	0.50
5	1	11	0.50
6	1	13	0.50
7	1	15	0.50
8	1	17	0.50
9	1	19	0.50
10	1	21	0.50
11	1	23	0.50
12	1	25	0.50
13	3	1	0.60
14	3	3	0.50
15	3	5	0.55
16	3	7	0.55
17	3	9	0.55
18	3	11	0.55
19	3	13	0.55
20	3	15	0.55
21	3	17	0.55
22	3	19	0.55
23	3	21	0.55
24	3	23	0.55
25	3	25	0.55
26	5	1	0.65
27	5	3	0.60
28	5	5	0.65
29	5	7	0.70
30	5	9	0.65
31	5	11	0.65
32	5	13	0.70
33	5	15	0.70
34	5	17	0.70
35	5	19	0.70
36	5	21	0.70
37	5	23	0.70
38	5	25	0.70

Classification using SVM with RBF kernel

	Num. Dimensions	Gamma	Accuracy
0	1	100	0.55
1	1	200	0.60
2	1	300	0.70
3	1	400	0.65
4	1	500	0.65
5	1	600	0.70
6	1	700	0.65
7	1	800	0.65
8	1	900	0.65
9	3	100	0.70
10	3	200	0.70
11	3	300	0.70
12	3	400	0.70
13	3	500	0.70
14	3	600	0.65
15	3	700	0.65
16	3	800	0.65
17	3	900	0.65
18	5	100	0.60
19	5	200	0.60
20	5	300	0.60
21	5	400	0.65
22	5	500	0.65
23	5	600	0.65
24	5	700	0.70
25	5	800	0.65
26	5	900	0.65

For the *k*-NN model, it is evident that keeping most of the components for PCA and assigning high values for the number neighbors yield the best result. For SVM though, there is no such clear trend. Almost any combination of parameters yield reasonable results for SVM.

By observing the results shown in section 2.5 and above, it is surprising to see how accurate the predictions are. However, these accuracies may be misleading because we are bound to have more rejected results than accepted results. For example, the selectivity of top schools are around 10% at best so the majority of data are those from rejected applicants. This means that if we were to build a predictor that simply outputs ‘reject’ every time, this model itself would produce significant accuracy. Hence, in our case, it is more meaningful to see whether the predictor accurately judged accepted applicants. Although the accuracy is not as impressive, both our *k*-NN model and SVM model correctly predicts 3 accepted students out of 8.

4. Conclusion

The main challenges we faced while implementing the features we described in this report were threefold. First of all, we had to cope with greatly varying structures of different websites. Secondly, our search results were sensitive to the accuracy of terms used in the query input by the user. That is, spelling errors or use of different synonyms affected the search results significantly. Thirdly, our analysis of past admission data for machine learning was limited by scarcity of high quality data.

The limitation imposed by first two challenges is that even though our results are close to 100% precision they suffer from low recall. As future work, we can make improvements on this by extracting more generalizable patterns to handle different websites and build a thesaurus of common query terms in this domain. To improve admission data analytics, we could include other popular admission sharing website that we are not yet aware of. Though one aspect of graduate school admission that is obviously clear from our analytics results is that high scores or research/work experience possession are not the sole factors of admission decision. The quality of papers published, statement of purpose, and recommendation letters all play a major role besides those factors. In any case, our predictor was implemented to give a rough advice to potential applicants and to let them gauge better of themselves.

Despite some of the limitations described above, our system has significant value in that it provides a single entry point for graduate application research – a feature which cannot be found elsewhere in the web. It eliminates the need to manually search and click on every single faculty member's homepage or current student's resume to look for relevant information as the summarization is presented automatically. Moreover, we include a website of different language (i.e. Korean) that many applicants would normally have restricted access to. In alignment with future web direction, our system has the potential to become a very specialized domain specific scout or gatherer.

5. References

[1] http://thegradcafe.com/

[2] http://www.gohackers.com/

[3] http://grad-schools.usnews.rankingsandreviews.com/best-graduate-schools

[4] http://www.indeed.com

[5] Christopher D. Manning, Prabhakar Raghavan and Hinrich Schuetze, *Introduction to Information Retrieval*, Cambridge University Press, 2008.

[6] Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

6. Appendix

To start our application, simply run 'GradInfo.py'. All required libraries are already installed in the *ugrad* machine.