

MAJOR TECHNICAL PROJECT ON

**BUILDING A SEARCH ENGINE FOR QUERYING
ACADEMIC SPECIALISTS**

INTERIM PROGRESS REPORT

to be submitted by

**SAHIL MUTNEJA
B11031**

*for the award of the degree
of*

**BACHELOR OF TECHNOLOGY IN
COMPUTER SCIENCE AND ENGINEERING**



**SCHOOL OF COMPUTING AND ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MANDI, MANDI**

APRIL 2015

1 Introduction

The problem here is to build a Search Engine that will provide the user with a list of experts when queried against a specialization. The experts that will be displayed will be from the premium Institutes(IITs, NITs) spanning across India. The list of professors will be such that the most relevant professors will be shown towards the top and so on in decreasing preference. The result will comprise of the links of web pages of the specialist along with the departments they belong in the respective institute.

The problem can be decomposed into three major subparts, written below, solving which will solve our problem.

- **Web Crawling** : The work of crawler is to recursively discover new and new pages, starting from the ones hard coded into the script, and filter the ones which are most appropriate for the purpose of search engine. It also makes sure that the spam pages are avoided when the crawler forward the pages to the Indexer.
- **Indexing** : In this phase, the forwarded content from the crawler is now analysed and gets saved in the data structure, where for each of the encountered word we will save the link of the page we saw the word in. Now, for each of the encountered word we will be having links of the pages the word is associated with. In addition to it, various other things are taken into consideration in order to improve the search results relevance.
- **Ranking and Searching** : This part deals with precomputing the ranks following the pagerank algorithm which takes into account the inlinks and outlinks of pages we have in our web graph. In addition to it there will be a number of other useful things that will be taken into ranking mechanism, for eg., frequency of word occurrence and much more. Once we have the ranks computed with us, which is done behind the scenes, we come to the part where user queries something. The queried phrase gets split into words and then looked up into the indexer where links get returned. These links will then get sorted on the basis of the ranks we have precomputed and finally gets displayed to the user.

Out of the things mentioned above, Crawling and Indexing are the things that are done offline. The output of the same are saved in a text file which will be later referred to by the ranker. The real time query as a result is quick and takes constant amount of time to return the links per query searched.

2 Objective and scope of the work

The project aims at building a search engine that performs specific sort of work on a big level. The problem with building a search engine on such a big level are the vast number of intricacies that comes with it. In order to meet with the requirements and also build a functional search engine we will be doing the following mentioned things that will provide a base for further

expansion of the project. The scope and limitations are as follows :

- As the title and introduction of the report tells us, we are building a search engine that is specific to our cause. The scope here is well defined, build a search engine that responds to the query in the form of specialization. The output will consist of the links of the concerned professors from the list of institutes hard coded into the system.
- The number of institutes that will be included will be around 5 to 6. Increasing the number of institutes requires large computation power and more memory, hence larger number of machines syncing together in a distributed environment.
- The search results will comprise only of the links of the concerned professors and faculty members. It will not consist of other different pages linked with the professors namely, their facebook pages, their social networking profiles and other academic stuff they are associated with. To include these as well, requires the crawler to crawl a large proportion of the web, which not only being not feasible but will also require a large computation power.
- As the number of pages crawled are limited, the conventional ranking mechanism will not be suitable for this case. In spite of using that and any other high scale ranking algorithm which requires a great amount of research and groundwork, we will be sticking to the inversion frequency algorithm and similar sort of works in order to rank them in order.

3 Work done in previous semester

The work done basically revolved around building a terminal based application where the search query is taken via the terminal and also the results are displayed on the terminal. Written below are the major things done in the last semester involving research and study of the existing search engines and building a basic prototype of the same.

- **Crawled Content** : Crawler work is to recursively visit websites that can be reached from a given website and maintain a record of it. One of the seed page is shown in Figure 1. It gets explored and all the anchor tags(links) present in the HTML of the page gets stored in the stack. As stack follows FIFO ordering, the links get selected from the stack and checked whether it satisfies the purpose of the project or not. If the link satisfies the constraints, it is taken into consideration else its discarded. The link taken into consideration again undergoes the similar thing, i.e., HTML content is explored and all the anchor tags are added into the stack. The popping operation of the stack keeps on going untill there are elements left in the stack.

Faculty

- [School of Computing and Electrical Engineering](#)
- [School of Basic Sciences](#)
- [School of Engineering](#)
- [School of Humanities and Social Sciences](#)

Figure 1: Seed page for IIT Mandi

In this phase of the work, the content that we were crawling is restricted to IIT Mandi and no other institutes were included. Adding more institutes will require more computation power of the system. The main focus out here was to build a basic system/prototype where we can see the power of searching some specialization on a small scale.

Towards the end of the crawler, we also updated a data structure(graph) of the data type list of lists. For each of the link we added to this graph all the outlinks that could be found on this page. This is used in the later part, where we will be ranking the pages via following the algorithm named as PageRank.

- **Indexing the Crawled Content** : In this stage the explored link content was captured and sent for processing. In this step, we used a dictionary data structure where the key is a string and the value consist of a list of links. These corresponding links will be such that this string(HTML line) is found in the content of the link. For the scale of IIT Mandi, the indexer consisted of 18,377 entries. At the time of the addition of a new string, it is firstly checked in the indexer. If the string already exists, the link of the page is added to the already created list against the keyword, else the new pair is created with key as the string and value as the list containing only one link.
- **Search Engine Running** : The search engine comprises of various scripts namely crawler, indexer and so on. In order to run the search engine, we executed the crawler script which in turn called the rest of the scripts and did all the computations. Once these computa-

tions are done, the indexer gets ready to feed the queries of the user. Everytime search engine needs to be used, the script was executed in order to make it functional.

- **Terminal Based Application** : The work done is based on the terminal based scripts as described above. The user enters the query in the terminal and is returned with the list of the links that are in accordance with the query. The links in this stage are not sorted according to their relevance, but only with respect to the random ordering they are encountered while doing the lookup operation in the Indexer. The same is illustrated in the Figures 2, 3 and 4. Figure 2 shows the search result for query "Machine Learning", Figure 3 shows the search result for query "Communication" and Figure 4 shows the search results for the query "Computer Networks".

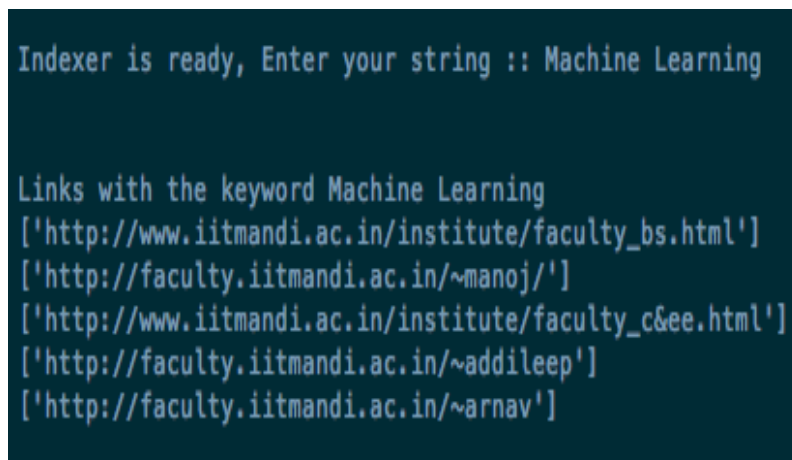
A terminal window with a dark blue background and light blue text. The text shows the prompt 'Indexer is ready, Enter your string :: Machine Learning' followed by a list of five URLs related to 'Machine Learning' at IIT Mandi. The URLs are: 'http://www.iitmandi.ac.in/institute/faculty_bs.html', 'http://faculty.iitmandi.ac.in/~manoj/', 'http://www.iitmandi.ac.in/institute/faculty_c&ee.html', 'http://faculty.iitmandi.ac.in/~addileep', and 'http://faculty.iitmandi.ac.in/~arnav'.

Figure 2: Results for search query "Machine Learning".

4 Work done from the previous evaluations to till today

From the last evaluation there are a number of significant changes done in the project. These changes range from ranking mechanism, optimization of the lookup operation, addition of new institutions and new interface for the users. Given below is the detailed explanation of all the mentioned above things.

- **Ranking Mechanism** : Till the last evaluation, we had with us the crawled content along with the indexer. On searching, we have given the links in a random order which is not in line with the model we propose. In order to have the ordering relevant, we need to add a ranking mechanism. The algorithm used here is PageRank. This algorithm analyses the inlinks and outlinks of every page and assigns a rank to it. These ranks are then sorted in decreasing order and finally displayed to the user. PageRank algorithm using mathematical equation is described below [2].

```

Indexer is ready, Enter your string :: Communication

Links with the keyword Communication
['http://faculty.iitmandi.ac.in/~varun/', 'http://faculty.iitmandi.ac.in/~varun']
['http://faculty.iitmandi.ac.in/~sudhir/']
['http://faculty.iitmandi.ac.in/~shekhar/']
['http://faculty.iitmandi.ac.in/~ajay/']
['http://faculty.iitmandi.ac.in/~vkn/']
['http://faculty.iitmandi.ac.in/~arti/']
['http://faculty.iitmandi.ac.in/~prasanth/']
['http://faculty.iitmandi.ac.in/~abhimanev/']
['http://faculty.iitmandi.ac.in/~achakraborty/']
['http://faculty.iitmandi.ac.in/~abbas/']
['http://www.iitmandi.ac.in/institute/faculty_c&ee.html']
['http://faculty.iitmandi.ac.in/~anand/']
['http://faculty.iitmandi.ac.in/~bsr/']
['http://faculty.iitmandi.ac.in/~padman/pubs.html']
['http://faculty.iitmandi.ac.in/~ramesho/']
['http://faculty.iitmandi.ac.in/~tag/']
['http://faculty.iitmandi.ac.in/~addileep']
['http://faculty.iitmandi.ac.in/~arnav']
['http://faculty.iitmandi.ac.in/~samar/']
['http://faculty.iitmandi.ac.in/~anil/']

```

Figure 3: Results for search query "Communication".

```

Indexer is ready, Enter your string :: Computer Networks

Links with the keyword Computer Networks
['http://faculty.iitmandi.ac.in/~arti/']
['http://www.iitmandi.ac.in/institute/faculty_c&ee.html']
['http://faculty.iitmandi.ac.in/~anand/']
['http://faculty.iitmandi.ac.in/~tag/']
['http://faculty.iitmandi.ac.in/~samar/']

```

Figure 4: Results for search query "Computer Networks".

Let x be a web page. Then

- $L(x)$ is the set of websites that link to x
- $C(y)$ is the out-degree of page y
- α is probability of random jump
- N is the total number of websites

$$PR(x) := \alpha \left(\frac{1}{N} \right) + (1 - \alpha) \sum_{y \in L(x)} \frac{PR(y)}{C(y)}$$

Note that the PageRanks form a probability distribution over web pages, so the sum of all web pages PageRanks will be one. PageRank or $PR(A)$ can be calculated using a simple iterative algorithm, and corresponds to the principal eigenvector of the normalized link matrix of the web. The page rank concept is well defined in the Figure 5.

In the figure, the alphabets are depicting pages or url and the lines depict the inlink and outlink from the pages. An arrow pointing from A to B means, in the content of page A there is an url that represents page B. The page rank states that if the number of inlinks of a page is more, its page rank will be higher. The percentage in the sphere represents the rank of the page. Greater the percentage of a page, more the page rank value and hence higher is the probability a user will want to access this page.

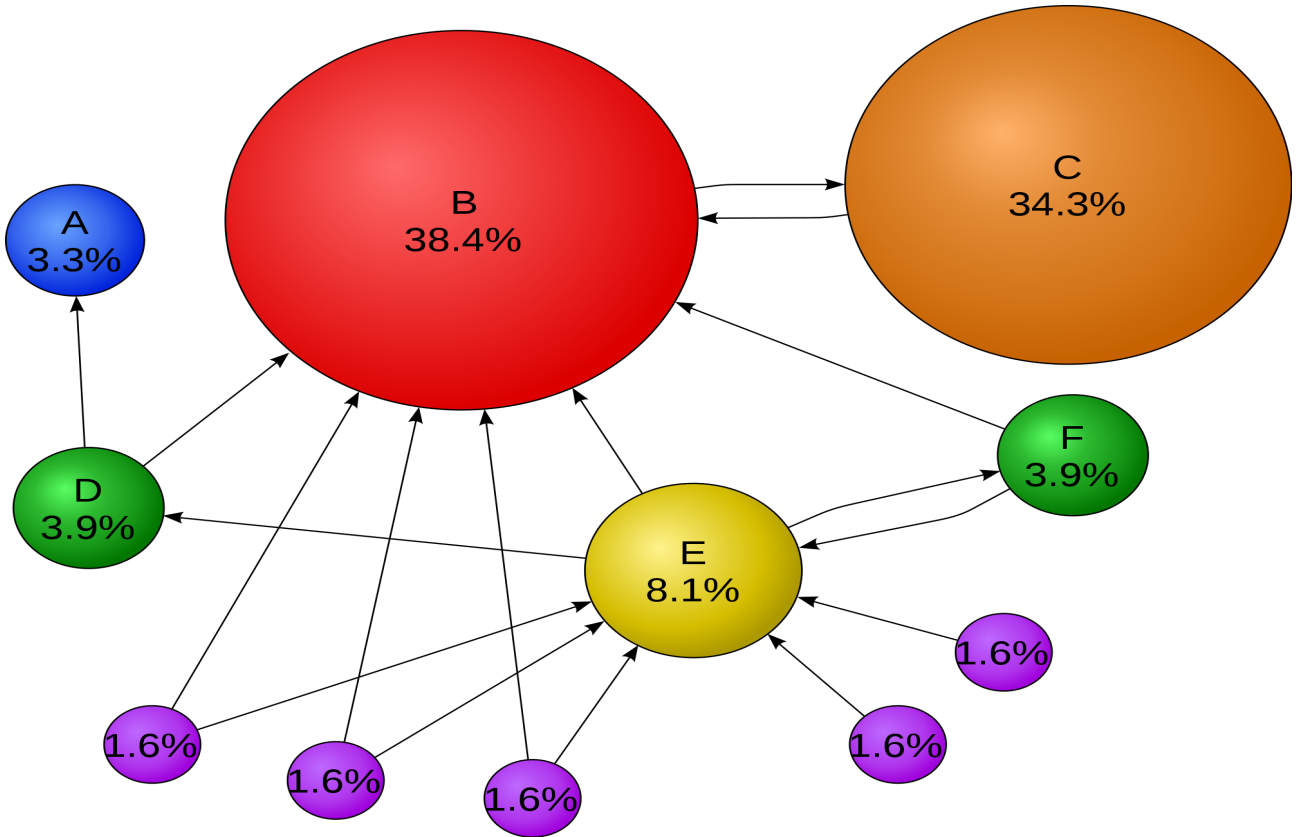


Figure 5: Illustration of PageRank

- **Addition of New Institutes :** In addition to IIT Mandi, now IIT Roorkee is also added to the list of the seed pages. In order to add the same, all the departments are added which interests our requirements. Addition of more institutes makes the indexer grow making our final results more useful. As of now, the number of institutes is restricted to a lower number so as to focus on the efficiency/quality of the results rather than the

quantity of the result. The Figure 6 shows the seed page corresponding to IIT Roorkee. This page acts as the base page which connects to the pages of every faculty member of IIT Roorkee.

- **Departments**
 - [Architecture and Planning](#)
 - [Applied Science and Engineering](#)
 - [Biotechnology](#)
 - [Chemical Engineering](#)
 - [Chemistry](#)
 - [Civil Engineering](#)
 - [Computer Science and Engineering](#)
 - [Earthquake Engineering](#)
 - [Earth Sciences](#)
 - [Electrical Engineering](#)
 - [Electronics and Communication Engineering](#)
 - [Humanities and Social Sciences](#)
 - [Hydrology](#)
 - [Management Studies](#)
 - [Mathematics](#)
 - [Mechanical and Industrial Engineering](#)
 - [Metallurgical and Materials Engineering](#)
 - [Paper Technology](#)
 - [Polymer and Process Engineering](#)
 - [Physics](#)
 - [Water Resources Development and Management](#)

Figure 6: Seed page for IIT Roorkee

- **Optimization of Lookup Operation :** In the initial version, indexer data structure consisted of the string as key which made the lookup operation complexity $O(n)$ which is linear and hence not feasible. In this case, for a given query we had to look into each of the key and search for the word presence in it resulting in the scan of entire dictionary. In this current version that we have, instead of line of HTML as key we included only the words that are valid. As a result of this, the lookup time became $O(1)$ which is constant. This is of prime importance as its the real time query that makes the search engine useful.

Adding to it the recomputation of the indexer is avoided by using a module in python named pickle [3], that allows the user to save any data structure computed in the script to a file on the disk. This allows to reuse the file again and again, hence saving a lot of time in recomputation of the same thing as the pages of the faculty will be undergoing almost no change in the span of days or may be months. In order to reflect the changes, a spider will be run in the background which will be crawling the web and changing the content

of the file if there is any change encountered in the websites of the concerned faculties. The spider is nothing but a bash script that gets run after a definite period of time, in our case it being 20 days. The work of script is to automatically redo the computations by executing the crawler and indexer program. The changes encountered via this in the faculty websites are thereafter updated in the pickle file mentioned above.

- **GUI for the Search Engine** : Web application has been created for the proper interactive working of the search engine. It provides the user a text box wherein keywords or phrases can be searched. The keywords are then split into words which are then searched and checked in the pickle file mentioned earlier. The results are then shown to the user in two different sets. The first set consists of the links corresponding to the faculty and professors web pages. The second set consists of the departments the result has been obtained from. The same is shown in Figure 7.

Enter the query you want to search :

Faculty Web Pages

- <http://faculty.jitmandi.ac.in/~ashok/>

A total of 1 results displayed

Department Web Pages

- http://www.jitr.ac.in/departments/HS/pages/People+Faculty_List.html
- http://www.jitr.ac.in/departments/HS/pages/About_the_Department_.html
- http://www.jitmandi.ac.in/institute/faculty_hss.html

A total of 3 results displayed

Figure 7: Searched result for query "Sociology"

- **Diagrams representing the working** : In order to make the back end of our search engine more clearer, a flow diagram of the same is shown in Figure 8. The figure tells us

about the various stages the query entered by the user undergoes till the final result in the form of links is displayed.

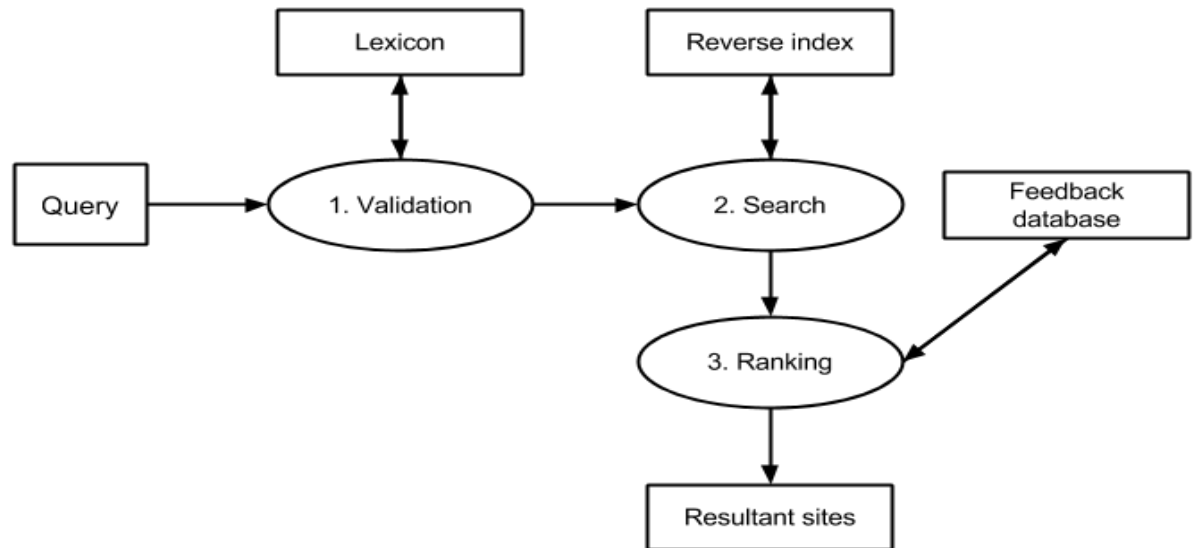


Figure 8: Flow diagram of search engine

The first step is query validation. The entered string is processed and only the relevant words are taken into consideration for further processing. The second step is the searcher. The words that are forwarded from the previous step are now looked up in the index data structure and the relevant links are loaded into the memory. The third step is the ranking. The links that we have in memory already have a rank associated with them. Now this pair of link and rank is sorted so that the link with the maximum rank is placed first and so on. The final step is to return the links in the order formed in the previous step.

In addition to the flow diagram, a use case diagram is shown in Figure 9. It tells us about the interaction between the users and different cases a user is involved in. As depicted in the figure, there are three types of user that deal with this search engine. The first one is the user that will access the engine. This user will submit a query which will be further processed and finally results will be displayed. The second type of user is the web developer. This user will write the scripts and will host the search engine on the web server platform to make it accessible to everyone. The third type of user is the admin. This user decides the functionalities, algorithms, technologies and the rules to be

followed while building the search engine.

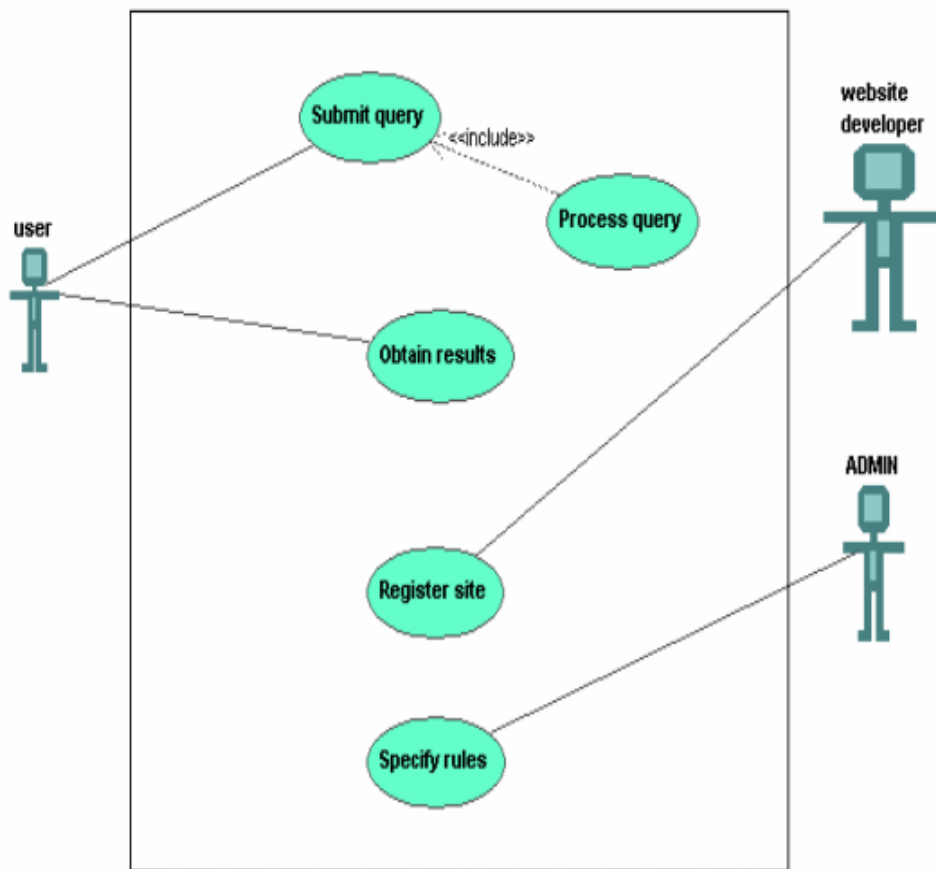


Figure 9: Use Case Diagram of Search Engine

5 Work intended to be done in future

As of now we have build a system along with the GUI that processes the query and gives back the useful and relevant links to the user. There are various aspects that still needs to be worked on in order to enhance the working and functionalities of the search engine. Some of the features that needs attention in the due course of the semester are listed below.

- **Ranking Techniques :** Currently the algorithm used is PageRank which is only suitable if the number of pages taken into consideration are large so that the number of inlinks and outlinks can reflect the correct rank. In our case, as we are crawling only a limited number of pages, some other algorithm or mechanism needs to be applied so that while displaying the links the most relevant link is displayed at the top. One of the thing that is currently used is to take into account the frequency of the words in

the document and use the logic that greater the frequency the more relevant is the page. Using this way and other techniques we can reach a point where the results are very close to the actual expected results. Researching and application of the new techniques is something that needs to be worked on.

- **Extending the Domain** : Once we have built a system that is giving good results on a small scale, the next target will be to expand the domain by including more number of institutes. As of now, we are crawling IIT Mandi and IIT Roorkee for the purpose of search engine. Next step will be to add more number of institutes to increase the usefulness of the search engine. The target is to involve about 5 to 6 IITs and make the results up to the mark.
- **GUI Enhancement** : This part deals with enhancing the existing application setup and make it more feature equipped and more user friendly.

The things mentioned above are the things once done will provide a base for further research and expansion on the built search engine. It can be extended to various dimensions and hence can be made more useful. Once the version that is targetted is built, the same will be uploaded on the server, so that anyone can access the built search engine and work on the code which will be open sourced. The same can be extended to a nation wide level once provided with the proper infrastructure in terms of hardware and software.

References

- [1] Bruce Croft, Donald Metzler, Trevor Strohman. *Search Engines: Information Retrieval in Practice*. Addison-Wesley, 2010 - Computers
- [2] Sergey Brin and Lawrence Page. *The Anatomy of a Large-Scale Hypertextual Web Search Engine*. Computer Science Department, Stanford University, Stanford, CA 94305
- [3] pickle Python object serialization *Documentation The Python Standard Library Data Persistence*. <https://docs.python.org/2/library/pickle.html>
- [4] Ian Rogers The Google Pagerank Algorithm and How It Works <http://www.cs.princeton.edu/~chazelle/courses/BIB/pagerank.htm>
- [5] Dave Evans *Build a Search Engine a Social Network*. Computer Science Department, University of Virginia