Building A Search Engine For Querying Academic Specialists

By Sahil Mutneja
under the mentorship of Dr. Dileep A.D.

Problem Statement

- Search Engine that provides the user with a list of experts when queried against a specialization
  - The experts will be from the premium Institutes(IIT’s, NIT’s) spanning across India
- The list will be such that the most relevant of the result will be towards the top.

Final Outcome

- The result will comprise of the links of web pages of the specialist along with the departments they belong in the respective institute.
- After analysing the crawled content, we will be building a web graph/network wherein nodes will be the specialization and the edges will be the links of the webpages or some relevant content

Description of the project

Decomposing the Problem Statement gives us three major subparts:

- Web Crawling
  - Browses and saves all the relevant web pages that can be found from the seed page we hard coded in our program.
- Indexing
  - The saved content gets decomposed and gets saved in the data structure based on the words occurrence
- Ranking and Searching
  - The pages will be ranked based on various parameters based on the domain of search engine.
  - The searched query will be decomposed and then searching and ranking of the words will give us the final result.

Plan as per the last review meeting

Work to be Completed	Date Assigned	Status of the Work
Build a Crawler	11th October 2014	Built a crawler that crawls the relevant content of IIT Mandi
Formatting via Parser	7th November 2014	Formed a Indexer wherein specialization lookup is supported.
Search Optimization Techniques	7th March 2015	Dictionary based implementation of Indexer to have all the operations in O(1) time
Implementation of ranker and Searcher for IIT Mandi	15th March 2015	Terminal based implementation of Search Engine on the scale of IIT Mandi
Extending to more number of Institutes	27th March 2015	Added IIT Roorkee to the crawler and indexer

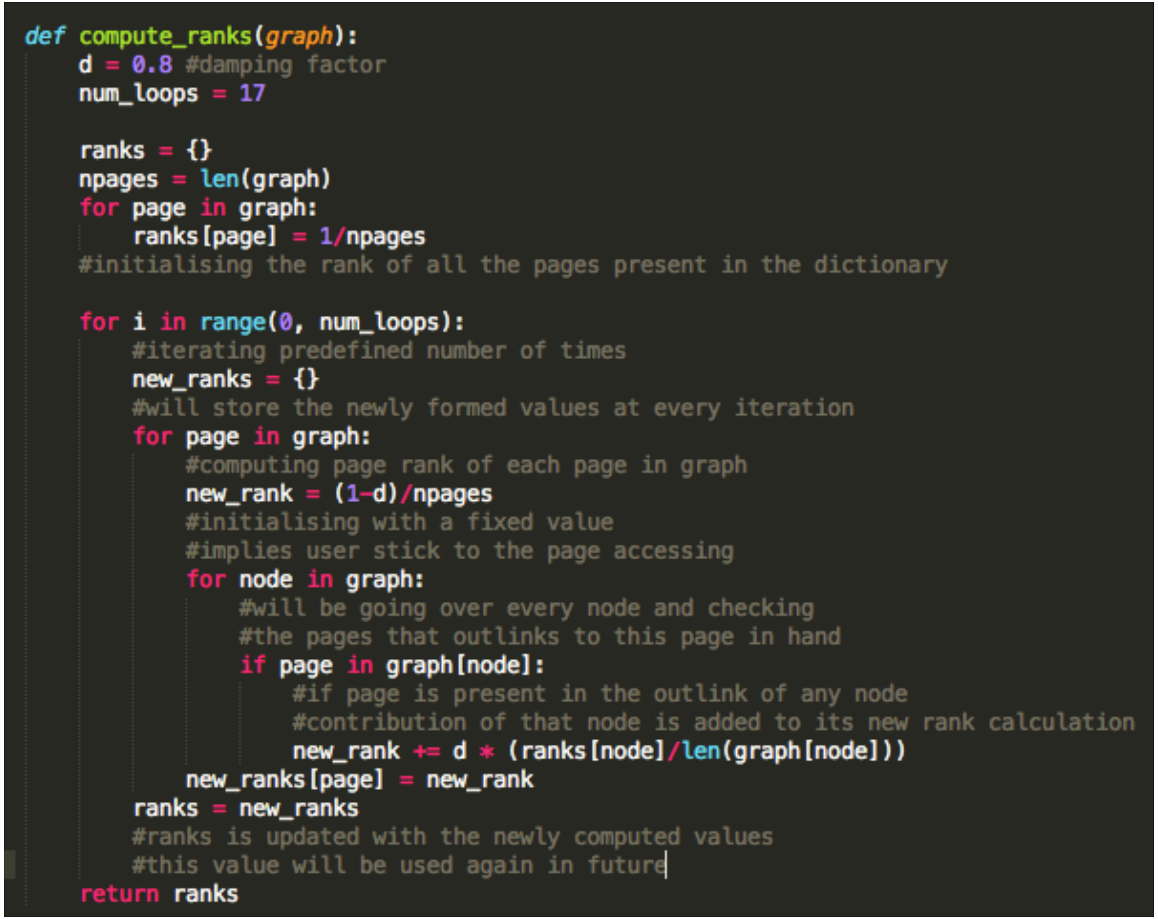
Significant Changes from the last meeting

The following additional things are added prior to what we had last time :

- Faster Lookup mechanism via Dictionary Implementation
- Used PageRank for the ranking mechanism
- Added IIT Roorkee to the set of institutes
- Built GUI for the search Engine, currently hosted on the local server

- Faster Lookup mechanism via Dictionary Implementation
  - Indexing done on the basis of words rather on the basis of line
  - Searched Query gets split into words and gets searched in the Indexer in O(1) time
- Used PageRank for the ranking mechanism
  - The mechanism initially used by google to rank pages
  - With the number of pages crawled and the number of inlinks and outlinks from one page to another, it gives a normalised rank to all the pages
- Added IIT Roorkee to the set of institutes
  - Relevant web pages of faculty and departments added to the crawler and indexer.
- Built GUI for the search Engine currently hosted on the local server
  - Built a webpage that given a query will look up into the already saved indexer and ranker and gets the user a list of web pages.

PageRank Algorithm Implementation



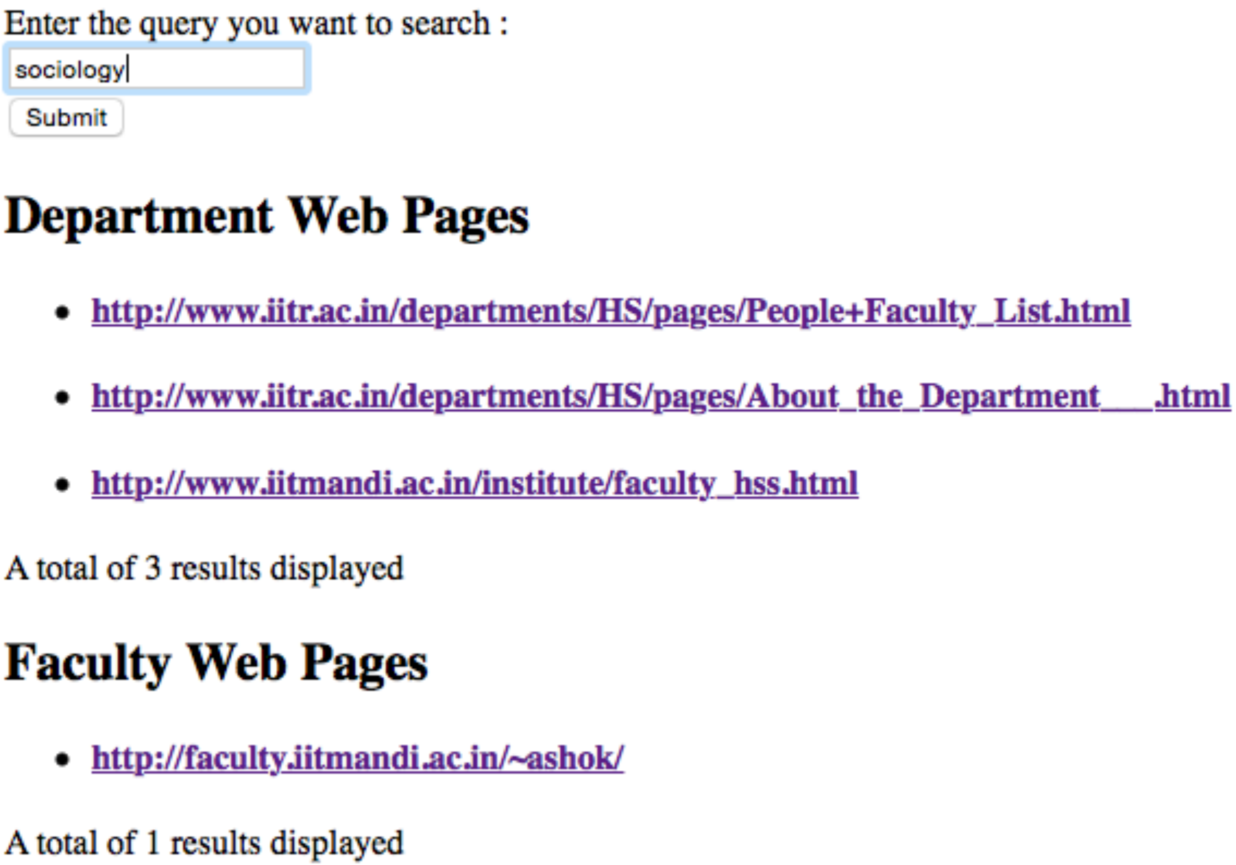
```python
def compute_ranks(graph):
    d = 0.8 #damping factor
    num_loops = 17

    ranks = {}
    npages = len(graph)
    for page in graph:
        ranks[page] = 1/npages
    #initialising the rank of all the pages present in the dictionary

    for i in range(0, num_loops):
        #iterating predefined number of times
        new_ranks = {}
        #will store the newly formed values at every iteration
        for page in graph:
            #computing page rank of each page in graph
            new_rank = (1-d)/npages
            #initialising with a fixed value
            #implies user stick to the page accessing
            for node in graph:
                #will be going over every node and checking
                #the pages that outlinks to this page in hand
                if page in graph[node]:
                    #if page is present in the outlink of any node
                    #contribution of that node is added to its new rank calculation
                    new_rank += d * (ranks[node]/len(graph[node]))
            new_ranks[page] = new_rank
        ranks = new_ranks
        #ranks is updated with the newly computed values
        #this value will be used again in future
    return ranks
```

Current status of project

- A basic GUI based search engine, with IIT Mandi and IIT Roorkee as its components.
- Rather than running the script over and over again, we are now saving the crawled and indexed information separately in a file using python module pickle, which can be later referenced to.
- The spider that crawls the web can be run in an automated script which will automatically change the content of the file and hence the search results.
- When a user searches for something, the query gets split into words and each word gets checked into the indexer giving us the links relevant to the search.
- The results comprises of two different sets, departments and the faculty.



Enter the query you want to search :

sociology

Department Web Pages

- http://www.iitr.ac.in/departments/HS/pages/People+Faculty_List.html
- http://www.iitr.ac.in/departments/HS/pages/About_the_Department___.html
- http://www.iitmandi.ac.in/institute/faculty_hss.html

A total of 3 results displayed

Faculty Web Pages

- http://faculty.iitmandi.ac.in/~ashok/

A total of 1 results displayed

Tasks to be accomplished

- Research on the ranking techniques. The PageRank technique used will give output in a not so accurate manner due to limited number of pages.
- Techniques needs to be devised that gives us the fetched pages which are relevant and according to our query.
- Various text analysis mechanism needs to be examined and implemented to get the proper ranking order.
- Improvements in GUI based search engine.
- Addition of more institutes to the domain of crawler and Indexer to make it usable at a small scale.
- Entire project to be hosted on web server to make it usable for anyone around the globe.

Course of Action with Time Frame (7th and 8th Sem included)

Work to be Completed	Tentative Date
Research on Text Classification, ML Algorithms for Indexer Enhancement and Algorithms for Ranker	In Winter Vacations
Search Optimization Techniques	7th March 2015
Implementation of Ranker and Searcher for IIT Mandi	15th March 2015
Extending domain of Crawler and Parser to more institutes and build Indexer	27th March 2015
Research on the techniques that could be used for the efficient working of the ranker.	15th April 2015
Implementation of the worked ranking techniques	27th April 2015
Addition of more number of Institutes	9th May 2015
Building Front End and Final Touches	17th May 2015